

AD A107490

10

LEVEL III

AP-E 950 176



TECHNICAL REPORT RG-CR-81-4

AUTOMATIC HANDOFF OF MULTIPLE TARGETS

J. S. Boland, III, H. S. Ranganath, and D. V. Satish Chandra  
Electrical Engineering Department  
Auburn University  
Auburn, Alabama 36830

14 September 1981

Prepared for:  
Guidance and Control Directorate  
US Army Missile Laboratory

DTIC  
ELECTE  
NOV 18 1981  
B



U.S. ARMY MISSILE COMMAND

Redstone Arsenal, Alabama 35898

Approved for public release; distribution unlimited.

DTIC FILE COPY

#### **DISPOSITION INSTRUCTIONS**

**DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT  
RETURN IT TO THE ORIGINATOR.**

#### **DISCLAIMER**

**THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN  
OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGN-  
ATED BY OTHER AUTHORIZED DOCUMENTS.**

#### **TRADE NAMES**

**USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES  
NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF  
THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-RG-CR-81-4	2. GOVT ACCESSION NO. AD-A107 490	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Automatic Handoff of Multiple Targets		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report-Phase I 10 Oct 80-14 Sept 81
7. AUTHOR(s) J. S. Boland, III, H. S. Ranganath and D. V. Satish Chandra		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Engineering Experiment Station Auburn University Auburn University, AL 36849		8. CONTRACT OR GRANT NUMBER(s) DAAH01-80-C-0258
11. CONTROLLING OFFICE NAME AND ADDRESS Headquarters, U.S. Army Missile Command ATTN: DRSMI-IYE/Cliver Redstone Arsenal, AL 35898		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 14 September 1981
		13. NUMBER OF PAGES 154
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Cleared for Public Release; Distribution Unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Correlation, Target Handoff, Image Correlation, Multiple Targets, Image Registration		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In order to fully utilize the potential of the "fire and forget" class of helicopter-borne missiles, it is necessary to solve the technical problems as- sociated with acquiring and handing off multiple targets from a precision point- ing and tracking system (PTS) to several missile seekers simultaneously or		

DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

almost so in a short period of time. The multiple target problem is that of locating targets and missile seeker aim points within the PTS field of view, deciding which target is to be assigned to each missile, generating error signals to the torquers in order to slew the missile LOS such that its assigned target is in the center of its FOV, and initiating automatic seeker tracking.

The task of locating a given smaller image within a larger image is known as "image registration". A detailed comparison of the important multiple image registration methods based on the number of arithmetic operations for software implementation and the complexity of hardware for real time implementation is presented. New methods of accomplishing multiple image registration which are computationally more efficient than the most commonly used template matching techniques (correlation and sequential similarity detection algorithm) are described. Several methods are compared with respect to hardware requirements and speed of computation. Conclusions and recommendations are given.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	iii
LIST OF TABLES . . . . .	vi
I. INTRODUCTION . . . . .	1
II. IMAGE REGISTRATION METHODS AND SIMULATION RESULTS . . . . .	5
A. Scenes Used for Simulation . . . . .	6
B. Binary Correlation Algorithm . . . . .	8
C. Image Registration Using Correlation of Adjacent Pixels . . . . .	10
D. Image Registration Based on Correlation of Adjacent Rows and Columns . . . . .	24
E. Image Registration Using Moments Method . . . . .	27
F. Image Registration Using Intraset and Inter-set Distances . . . . .	29
G. A New Feature Matching Image Registration Technique . . . . .	32
1. Derivation of the Computational Requirements for Multiple Image Registration . . . . .	36
2. Simulation Results . . . . .	40
H. Image Registration Using Haar Coefficients . . . . .	50
1. Modified and Rationalized Haar Transforms . . . . .	57
2. Selection of Haar Coefficients as Features . . . . .	61
3. Derivation of the Computational Requirements for Multiple Image Registration . . . . .	68
4. Simulation of Haar Transforms . . . . .	75
I. Digital Image Registration Using Walsh/Hadamard Transform . . . . .	86
III. HARDWARE IMPLEMENTATION . . . . .	95
A. Standard Cross Correlation Algorithm . . . . .	95
B. Binary Correlation Algorithm . . . . .	98
C. Correlation of Adjacent Pixels Method . . . . .	98
1. Row Feature Extractor . . . . .	102
2. Column Feature Extractor . . . . .	106

3.	Real-time Implementation . . . . .	108
4.	Matching of Feature Vectors . . . . .	110
D.	Hardware Requirements of the Standard Correlation and Correlation of Adjacent Pixels Method . . . . .	112
E.	New Feature Matching Method . . . . .	114
1.	Row Feature Extractor . . . . .	114
2.	Column Feature Extractor . . . . .	117
3.	Real-time Hardware Implementation of the New Feature Matching Method . . . . .	119
4.	Multiple Image Registration . . . . .	120
5.	Hardware Requirements of the New Feature Matching Method . . . . .	120
F.	Haar Transform Method . . . . .	121
1.	Pipelined Implementation of the Haar Transform . . . . .	123
2.	Parallel Implementation of the Haar Transform . . . . .	125
3.	Real-time Hardware Implementation of the Haar Transform Method . . . . .	125
4.	Multiple Image Registration . . . . .	128
5.	Hardware Requirements of the Haar Transform Method . . . . .	129
IV.	HARDWARE vs. TIME TRADEOFFS FOR MULTIPLE TARGET HANDOFF PROBLEM . . . . .	131
A.	Method One . . . . .	131
B.	Method Two . . . . .	135
C.	Method Three . . . . .	137
D.	Method Four . . . . .	138
E.	Method Five . . . . .	139
F.	Method Six . . . . .	139
G.	Method Seven . . . . .	141
V.	CONCLUSIONS AND RECOMMENDATIONS . . . . .	147
	REFERENCES . . . . .	153

## LIST OF FIGURES

1-1. Helicopter missile fire control system . . . . .	2
1-2. Search area and windows . . . . .	3
2-1. Number of additions to register n windows . . . . .	13
2-2. Number of multiplications to register n windows . . . . .	14
2-3. Number of equivalent additions to register n windows . . . . .	15
2-4. Layout for preprocessing the reduced high resolution image . .	23
2-5. Number of additions to register n windows . . . . .	41
2-6. Number of multiplications to register n windows . . . . .	42
2-7. Number of equivalent additions to register n windows . . . . .	43
2-8. An 8 x 8 Haar matrix . . . . .	51
2-9. Basis plane for an 8 x 8 Haar transform . . . . .	53
2-10. An 8 x 8 spatial domain digital image . . . . .	54
2-11. The (3, 4) basis plane for Figure 2-9 . . . . .	55
2-12. The weighting factors for the 8 x 8 transform-domain image .	56
2-13. A flow diagram for an 8 point fast Haar transform (FHT) . . .	58
2-14. An 8 x 8 modified-Haar matrix . . . . .	59
2-15. An 8 x 8 rationalized-Haar matrix . . . . .	59
2-16. Transform coefficient variance distribution for low resolution FLIR images . . . . .	63
2-17. Location of the 64 Haar transform coefficients having largest variances (Low resolution FLIR images) . . . . .	64
2-18. Location of the 64 modified Haar transform coefficients having largest variances (Low resolution FLIR images) . . . . .	65

2-19.	Location of the 64 rationalized Haar transform coefficients having largest variances (Low resolution FLIR images) . . . . .	66
2-20.	Transform coefficient variance distribution for high resolution FLIR images . . . . .	67
2-21.	Location of the 64 Haar transform coefficients having largest variances (High resolution FLIR images) . . . . .	69
2-22.	Location of the 64 modified Haar transform coefficients having largest variances (High resolution FLIR images) . . . . .	70
2-23.	Location of the 64 rationalized Haar transform coefficients having largest variances (High resolution FLIR images) . . . . .	71
2-24.	Number of additions to register n windows . . . . .	73
2-25.	Number of multiplications to register n windows . . . . .	74
2-26.	Number of equivalent additions to register n windows . . . . .	76
2-27.	Sequency ordered 8 x 8 Walsh/Hadamard matrix . . . . .	87
2-28.	Two dimensional basis plane for an 8 x 8 Walsh/Hadamard transform . . . . .	88
2-29.	Comparison of the arithmetic requirements of various image registration algorithms . . . . .	93
3-1.	Schematic for correlation method . . . . .	97
3-2.	Schematic for the binary correlation algorithm . . . . .	99
3-3.	KxN array of shift registers containing the first K rows of the search area of size MxN at the end of KN sampling periods . . . . .	101
3-4.	Row feature extractor . . . . .	103
3-5.	Contents of shift registers in row K at the end of KN-1 sampling periods . . . . .	104
3-6.	Column feature extractor . . . . .	107
3-7.	Real-time implementation of correlation of adjacent pixels method and the new feature matching method . . . . .	109
3-8.	Correlation of feature vectors . . . . .	111
3-9.	Hardware implementation of row-feature extractor . . . . .	115
3-10.	Hardware implementation of column-feature extractor . . . . .	118



3-11.	Pipelined implementation of the 8-point Haar transform . . .	.124
3-12.	Parallel implementation of the 8-point Haar transform . . .	.126
3-13.	Real-time implementation of the feature matching method based on Haar coefficients . . . . .	.127
4-1.	Block diagram for multiple target handoff methods 1, 2, and 3 . . . . .	.132
4-2.	Block diagram for feature matching multitarget handoff . . .	.140
4-3.	Automatic target recognizer block diagram . . . . .	.142
4-4.	Block diagram of combined target recognizer and feature matching multitarget handoff hardware . . . . .	.143

Accession For	
NTIS CHAS1	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

## LIST OF TABLES

1. Distance between the targets and the sensors . . . . .	6
2. Binary correlation simulation results . . . . .	11
3. Correlation simulation results . . . . .	12
4. Registration simulation results for correlation of adjacent pixels method . . . . .	20
5. Image registration using correlation of adjacent pixels . . . . .	25
6. Summary of results in Table 4 and Table 5 . . . . .	28
7. Simulation results of intraset and intersets distance methods . . . . .	33
8. Summary of intraset and intersets simulation results presented in Table 7 . . . . .	34
9. Simulation results of new feature matching image registration technique . . . . .	48
10. Summary of results in Table 9 . . . . .	49
11. Computational requirements of various two-dimensional transforms . . . . .	60
12. Simulation results of the feature matching image registration technique using Haar coefficients . . . . .	82
13. Summary of results in Table 12 . . . . .	84
14. Simulation results of the feature matching technique using WHT and Haar transform coefficients . . . . .	90
15. Summary of results in Table 14 . . . . .	91
16. Hardware requirements of the standard correlation and correlation of adjacent pixels methods . . . . .	113
17. Hardware requirements of the new feature matching method . . . . .	122
18. Hardware requirements of the feature matching method using Haar coefficients . . . . .	130

19. Firing time for seven methods . . . . .	145
---	-----

## I. INTRODUCTION

One concept which could increase the firepower and survivability of the "fire-and-forget" class of helicopter borne missile systems would be to acquire and hand off simultaneously, or almost so, multiple targets from a precision pointing and tracking system (PTS) to several missile seekers and fire in a very short period of time. A typical over-all fire control configuration is shown in Figure 1-1. The pointing and tracking system typically consists of an optics train, line of sight (LOS) stabilization system, forward looking infrared (FLIR) imaging system, manual and autotrack system, laser range finder and associated electronics. An imaging missile seeker could be an infrared type. It is assumed that during preflight checkout or during the actual flight, the lines of sight of all the missile seekers are aligned with the line of sight of the PTS. However, due to gyro drift, boresighting inaccuracies, vehicle vibration and flexure, etc., the seekers will not remain boresighted with the PTS. Since the PTS has a larger field of view (FOV) in both axes than the missile seekers, it is expected that the FOV of all the missile seekers will be located within the FOV of the PTS. The multiple target problem then becomes that of locating missile seeker LOS aim points within the PTS field of view, deciding which target is to be assigned to each missile, generating error signals to the torquers in order to slew each missile LOS such that its assigned target is in the center of its FOV, and initiating automatic seeker tracking. The work reported in this final report

assumes that some method is available for acquiring, recognizing, and marking potential targets within the FLIR FOV (eg., MTI radar and/or a target recognizer).

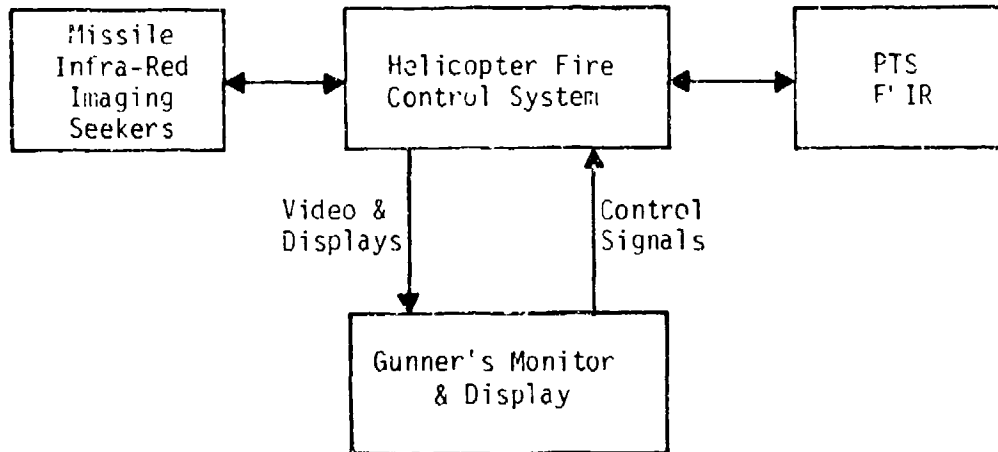


Figure 1-1. Helicopter missile fire control system.

In general, the task of locating a given smaller image within a larger image is known as "image registration". The smaller image is referred to as the window or the reference and the larger image is called the search area. Using the above notation with the multiple target problem, the low resolution image obtained from the PTS sensor is the search area and the high resolution image obtained from each missile seeker is a window. Therefore, there is one search area and more than one, say  $n$ , windows. It is assumed that all the  $n$  windows are completely located within the search area. Now the problem of multiple image registration can be defined as that of finding  $n$  subimages within the search area which best match the  $n$  windows. The  $n$  windows  $W_1, W_2, \dots, W_n$  and the

search area  $S$  are shown in Figure 1-2.  $S$  is an  $M \times N$  array of digital picture elements which may assume one of  $G$  possible levels on the gray scale.

$$0 \leq S(i,j) \leq G-1, \quad (1-1)$$

for  $1 \leq i \leq M, 1 \leq j \leq N$ .

$W_k$  is a  $K \times L$  array of pixels having the same gray scale range.

$$0 \leq W_k(\ell,m) \leq G-1, \quad (1-2)$$

for  $1 \leq \ell \leq K, 1 \leq m \leq L$  and  $k = 1, 2, \dots, n$ .

Each  $K \times L$  subimage of  $S$  can be uniquely identified by its upper left corner's coordinates. Let  $S_{i,j}$  denote the  $K \times L$  subimage of  $S$  whose upper left corner is  $(i,j)$ .

$$S_{i,j}(\ell,m) = S(i+\ell-1, j+m-1), \quad (1-3)$$

for  $1 \leq \ell \leq K, 1 \leq m \leq L$ ,

and  $1 \leq i \leq M-K+1, 1 \leq j \leq N-L+1$ .

If  $S$  and  $W_k$  do not differ in pixel resolution and rotation, the multiple image registration problem reduces to that of finding  $(i_k^*, j_k^*)$  such that  $S_{i_k^*, j_k^*}$  best matches  $W_k$ , for  $k = 1, 2, \dots, n$ .

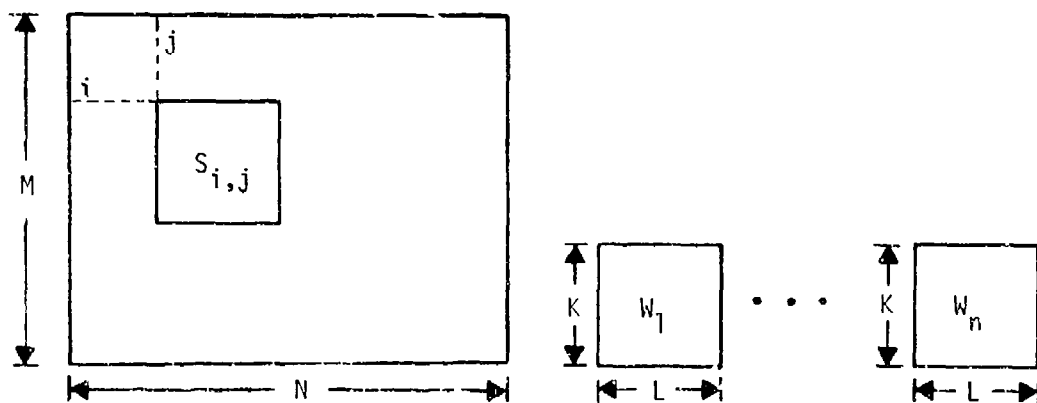


Figure 1-2. Search area and windows.

Several template matching and feature matching methods of accomplishing digital image registration are presented in detail in the Final Technical Report of Phase I of contract DAAH01-80-C-0258. It has been shown that, in general, feature matching algorithms require fewer arithmetic operations for software implementation and less hardware for real-time hardware implementation than the template matching methods such as "correlation" and "sequential similarity detection algorithms", if the number of windows is sufficiently large. In Chapter II, the results of simulation of the important registration methods in [1] are given and their registration accuracies are compared. Two new feature matching methods, one based on the Haar transform and the other based on adjacent pixel differences along the rows and columns of a digital image, are also presented in Chapter II. Possible hardware implementations of the algorithms discussed in Chapter II are given in Chapter III.

The main thrust of this effort has been to determine the reliability and accuracy of existing feature matching methods and to define new methods. It was felt that in order to reduce computation time and/or hardware requirements, the actual matching process should be based on a few dominant features rather than image gray level or edge content. The hardware vs. time tradeoffs for several sequential correlation image matching techniques and for the feature matching technique are discussed in Chapter IV. The final technique in Chapter IV suggests a technology program for developing combined target recognizer and feature matching handoff hardware. The conclusions and recommendations are given in Chapter V.

## II. IMAGE REGISTRATION METHODS AND SIMULATION RESULTS

The problem of registering a given image within a larger image uses techniques fundamental to the disciplines of image processing and pattern recognition. The image registration algorithms are broadly classified as template matching algorithms and feature matching algorithms. Template matching methods such as "correlation" and "sequential similarity detection algorithms" are widely used for the determination of local similarity between two images in the single image registration problem. For multiple image registration, it has been shown that feature matching methods are computationally more efficient than template matching methods if the number of windows is sufficiently large [1]. This is because the amount of computation for template matching algorithms is directly proportional to the number of windows whereas in feature matching algorithms it is not. In feature matching algorithms, features are extracted for all subimages of the search area and windows only once and the matching procedure is repeated once for each window. Since the amount of computation required to match the features is negligible compared to that required to compute the features, the computational efficiency of the feature matching methods increases with the number of windows.

Three possible feature matching methods of accomplishing multiple image registration based on image moments, intraset and interset distances, and correlation of adjacent pixels are described in detail in the Final Technical Report of Phase I of contract DAAH01-80-C-0258. All three



methods were shown to be computationally more efficient than the most widely used cross correlation method if the number of windows is sufficiently large. However, the most important performance measure of an algorithm is its registration accuracy. During this phase of the contract, an effort was made to determine the accuracy of the above feature matching methods through simulation. In addition to the above methods, two new feature matching methods, one based on the Haar transform and the other based on adjacent pixel differences along the rows and columns of a digital image are also presented in this chapter. The performance (registration reliability) of each method is compared with that of a binary correlation algorithm.

#### A. Scenes Used for Simulation

Four different sets of images for targets at four different ranges are used for simulation. Each set consists of a low resolution FLIR image and four high resolution FLIR images (i.e., four targets at each range). The fields of view of the high resolution images are completely located within the field of view of the low resolution image. The ranges to the targets located at the four different positions are given in Table 1.

Table 1. Distance between the targets and the sensors.

Position	Distance
A	3 km
B	3.5 km
C	4 km
D	5 km

All image registration methods presented in this report are based on the assumption that the search area and windows do not differ in rotation and spatial pixel resolution. However, due to the difference in sensor characteristics the low resolution FLIR image (search area) and high resolution FLIR images (windows) do not have the same spatial pixel resolution. In order to equalize the spatial pixel resolution, the images are preprocessed as described below.

1. Each field of the unprocessed high resolution FLIR image is a  $406 \times 392$  array of digital picture elements which can assume one of 256 possible levels on the gray scale. The horizontal and vertical scale factors which are used to reduce the pixel resolution of high resolution FLIR images are predetermined to be 5.0 and 4.108, respectively. Reducing the resolution using the above scale factors yields an array of size  $98 \times 78$ .
2. Each field of unprocessed low resolution FLIR imagery contains  $406 \times 392$  pixels where each pixel can assume one of 256 possible gray levels. The horizontal and vertical scale factors which reduce the resolution of the low resolution FLIR image to that of the reduced high resolution image are predetermined to be 1.25 and 1.0, respectively. Reducing the resolution using the above scale factors yields a reduced low resolution FLIR image of size  $406 \times 313$ . The 1.25 horizontal scale is necessary in order to reduce the effective sampling rate for the digitized video from 12.5 MHz to 10 MHz.

The above preprocessed images are inputs to each of the image registration algorithms simulated.

### B. Binary Correlation Algorithm

Cross correlation is the most widely used method of accomplishing digital image registration. If the window and search area are preprocessed and transformed into binary images to accomplish image registration, a substantial data reduction and savings in computation can be achieved. Use of binary images not only reduces the computational burden of the correlation algorithm for software implementation, but also reduces the complexity of a real-time hardware implementation. A correlation algorithm which is easily implementable in real-time using simple digital hardware (exclusive-NOR gates) is given in Reference [2]. Simulation results given in References [3, 4] give the accuracy trade-offs for binary correlation. Because of the previous analysis and hardware design, the binary correlator is used as a basis for comparison of the performance of feature matching registration algorithms in this report. The registration process using binary images is described in the following steps.

- Step 1: A  $K \times L$  reference image  $W$  is selected from the reduced high resolution FLIR image.
- Step 2: The mean pixel value  $\mu_W$  of  $W$  is computed. The reference is then transformed to a binary image by quantizing pixels with values greater than or equal to  $\mu_W$  to ones and those with values less than  $\mu_W$  to zeros.
- Step 3: Each  $K \times L$  subimage of the search area (preprocessed low resolution FLIR image) is also quantized to two levels in the same manner as the reference image.

Step 4: The binary images are then correlated. The correlation function,  $R(i,j)$ , which is a measure of similarity between  $W$  and  $S_{i,j}$ , is computed as

$$R(i,j) = KL - \sum_{l=1}^K \sum_{m=1}^L |S_{i,j}(l,m) - W(l,m)| \quad (2-1)$$

for  $1 \leq i \leq M-K+1$  and  $1 \leq j \leq N-L+1$ .

Maximum similarity is expected at the registration point.

The four preprocessed low resolution images and sixteen high resolution images described in a previous section are used for simulation. For each scene used for simulation, the following three cases are considered.

Case 1: Reference images of size  $32 \times 32$  are chosen to include the prominent features in the reduced high resolution FLIR images. In other words, references are not necessarily taken from the center of the field of view of the reduced high resolution image. For this case the range values given in Table 1 are meaningless since those ranges refer to the target in the center of the field of view.

Case 2: Reference images of size  $32 \times 32$  are chosen from the center of the fields of view of the reduced high resolution images.

Case 3: Reference images of size  $32 \times 64$  are chosen from the center of the fields of view of the reduced high resolution images. For this case, one of the reference images (position A, target #1) is not completely located within the search area and therefore only fifteen scenes are simulated.

Correlation simulation results for each of the above cases are given in detail in the monthly reports of December 1980 through June 1981 and

are not given in this report. A summary of results is presented in Table 2. In Table 2, a "1" indicates that the true registration point appeared as the first maximum in the correlation surface. Similarly, a "2", "3", or "4" indicate that the registration point appeared as second, third or fourth maximum, respectively. An "X" indicates that the registration point was not within the first four maxima. The first four maxima in the correlation surface are found by finding the maximum value first. The second maximum is found by masking out an array of  $21 \times 21$  pixels centered about the first maximum and then searching the remaining correlation surface for its maximum value. Third and fourth maxima are found similarly. For Case 1 the true registration point appeared as the first maximum in 9 out of 16 cases, was within the first four maxima for 3 additional cases, and was not within the first four maxima for the remaining 4 cases. These results and the results for Cases 2 and 3 can be found in Table 3. When reference images were chosen to include prominent features in the reduced high resolution image, the performance of the binary correlation algorithm was satisfactory. For the other two cases, the method's performance was poor.

### C. Image Registration Using Correlation of Adjacent Pixels

A method of accomplishing digital image registration based on correlation of adjacent pixels is given in Reference 1. In Reference 1, this method was shown to be computationally more efficient than any of the template matching or feature matching algorithms studied to-date. Multiplication and addition requirements for this method and other important registration methods are derived in [1]. The number of additions

Table 2. Binary correlation simulation results.

Scene		Case 1	Case 2	Case 3
Position A	Target #1	1	X	-
	Target #2	1	X	X
	Target #3	X	X	X
	Target #4	X	X	X
Position B	Target #1	X	X	X
	Target #2	1	1	3
	Target #3	1	X	X
	Target #4	2	X	X
Position C	Target #1	2	1	1
	Target #2	X	X	3
	Target #3	1	1	1
	Target #4	1	2	X
Position D	Target #1	1	X	X
	Target #2	1	X	X
	Target #3	1	X	X
	Target #4	4	X	X

Table 3. Correlation simulation results.

	Registration point is first maximum	Registration point 2nd, 3rd, or 4th maximum	Registration point not in four maxima
Case 1	9	3	4
Case 2	3	1	12
Case 3	2	2	11

and multiplications required by various registration methods as a function of the number of windows of size  $32 \times 32$  within a search area of size  $240 \times 256$  are shown in Figures 2-1 and 2-2, respectively. The equivalent number of additions required by each method as a function of the number of windows is given in Figure 2-3. Figure 2-3 is derived from Figures 2-1 and 2-2 assuming that each real multiplication is equivalent to three real additions. From the above figures it is clear that, even for single image registration, the method based on correlation of adjacent pixels requires fewer arithmetic operations than the standard cross correlation algorithm. Since the feature vector of the subimage  $S_{i+1,j}$  or  $S_{i,j+1}$  can be computed from the feature vector of the subimage  $S_{i,j}$  with very few arithmetic operations, this method is very promising for real-time implementation. In order to determine the reliability and accuracy of the correlation of adjacent pixels method, simulation was conducted both with and without intensity level normalization of images. The registration technique and simulation results are presented next for all cases considered.

Case 1: In Case 1, an attempt is made to accomplish image registration without intensity level normalization or any other preprocessing

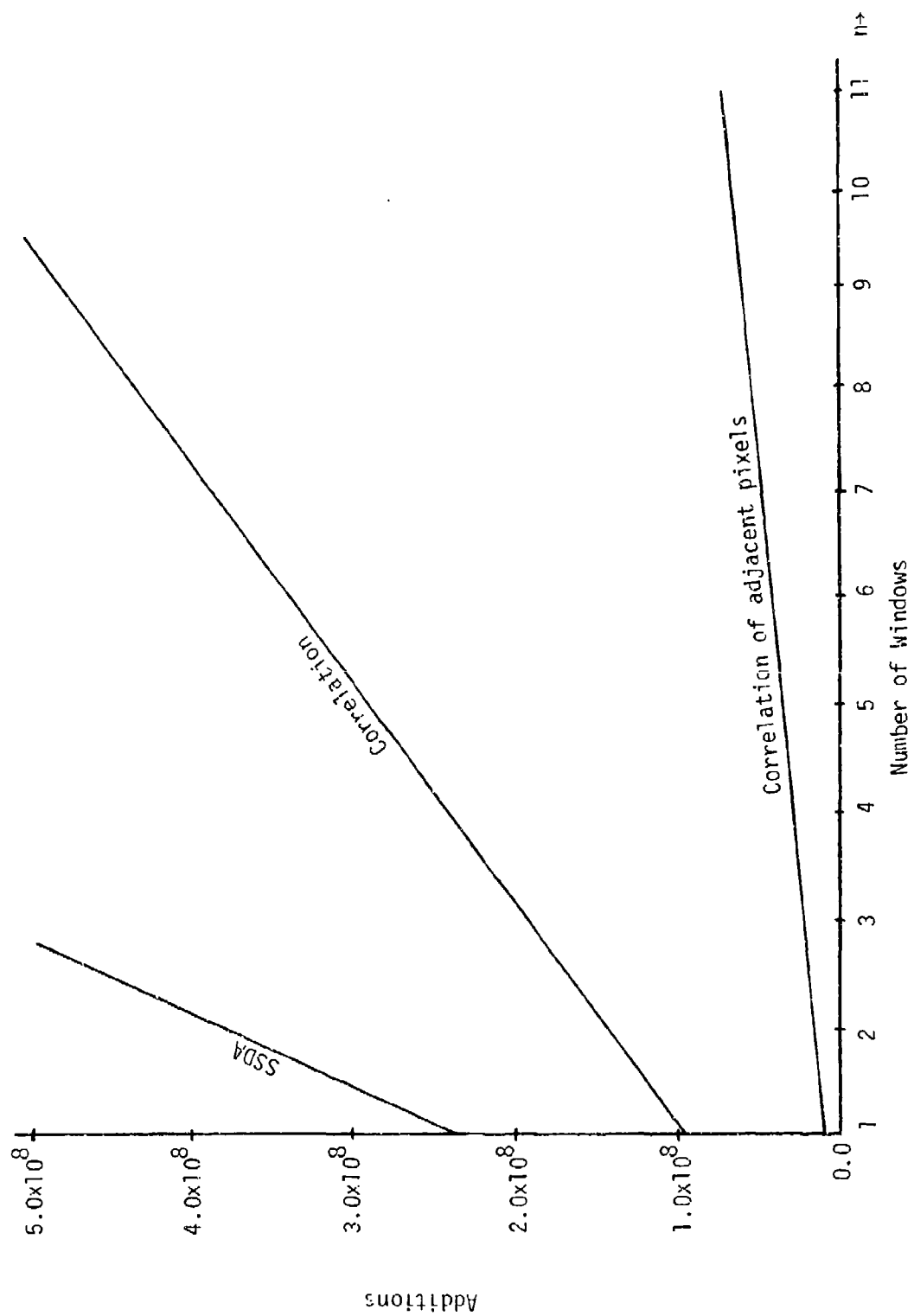


Figure 2-1. Number of additions to register  $n$  windows.



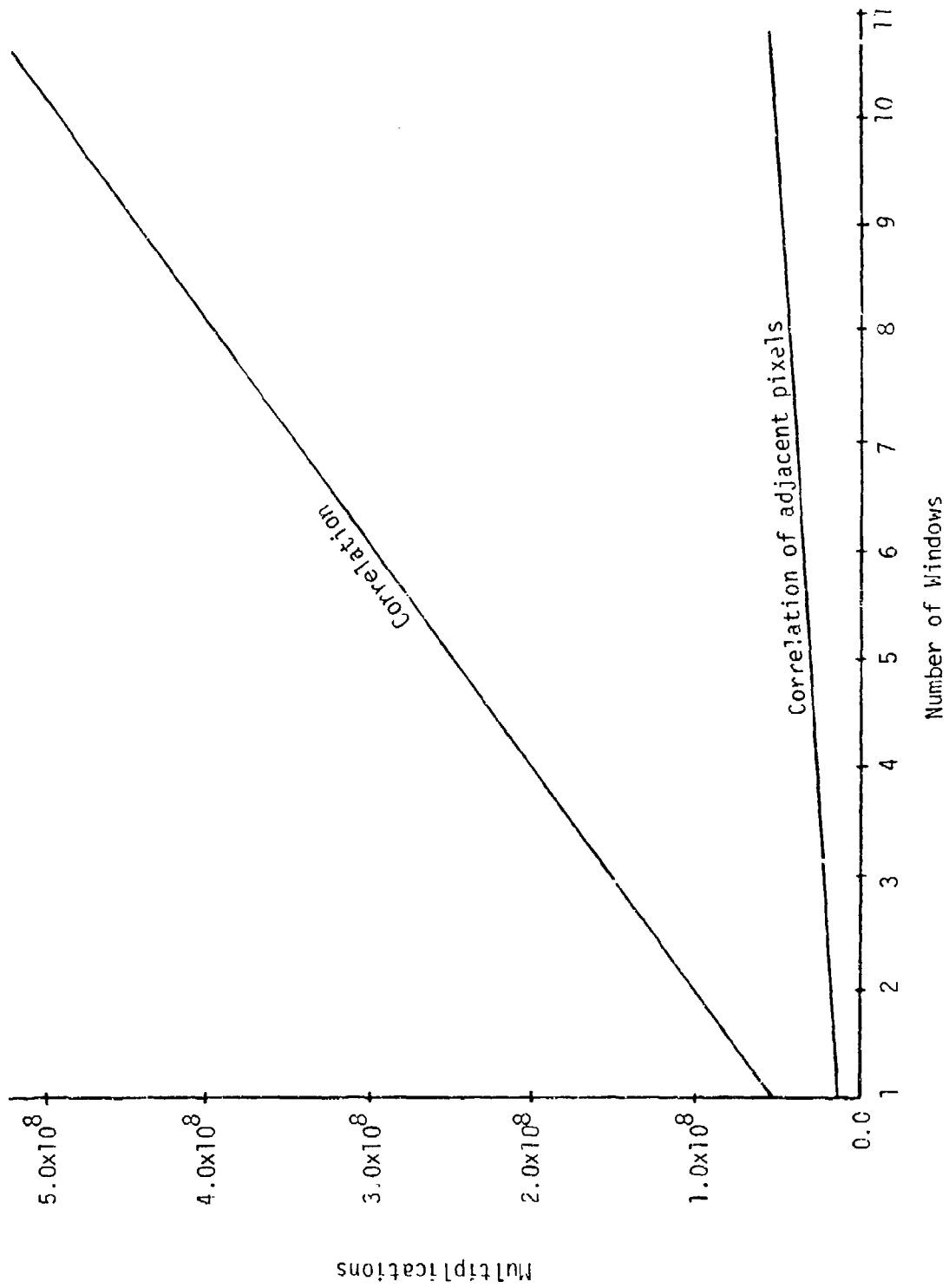


Figure 2-2. Number of multiplications to register  $n$  windows.

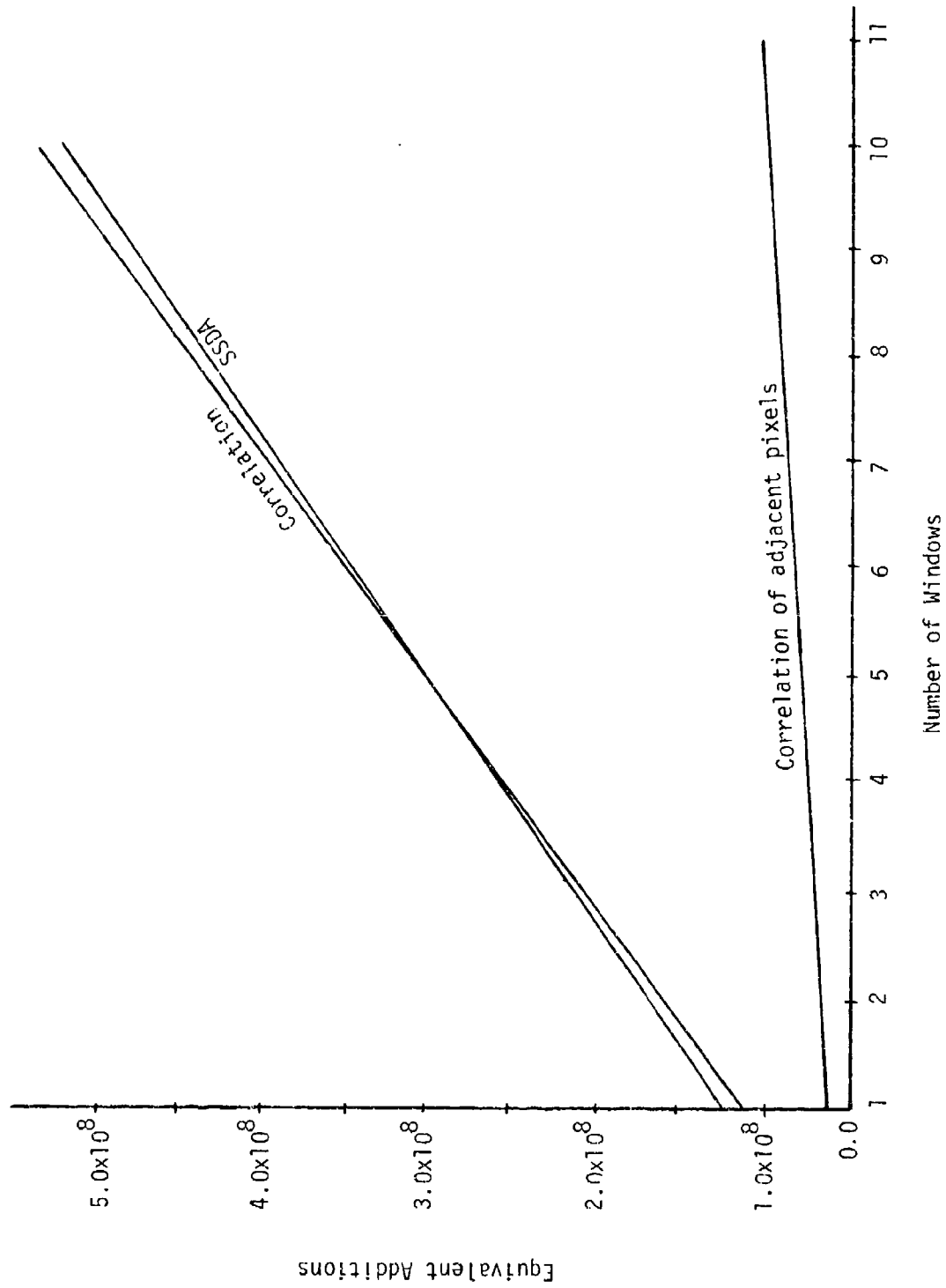


Figure 2-3. Number of equivalent additions to register n windows.

of images. The window and search area whose elements can assume one of 256 levels on the gray scale are used with the registration procedure as given in the seven steps below.

Step 1: A reference image  $W$  of size  $K \times L$  is taken from the reduced high resolution FLIR image to include prominent features of the image.

Step 2: The normalized correlation between adjacent pixels of each row is computed as

$$\rho_{\ell} = \frac{\sum_{m=1}^{L-1} W(\ell, m) W(\ell, m+1)}{\sum_{m=1}^{L-1} W^2(\ell, m)}, \quad (2-2)$$

for  $\ell = 1, 2, 3, \dots, K$ .

Step 3: The normalized correlation between the adjacent pixels of each column is computed as,

$$\sigma_m = \frac{\sum_{\ell=1}^{K-1} W(\ell, m) W(\ell+1, m)}{\sum_{\ell=1}^{K-1} W^2(\ell, m)}, \quad (2-3)$$

for  $m = 1, 2, 3, \dots, L$ .

Step 4: The  $(K+L)$ -dimensional feature vector,  $V_W$ , of the window  $W$  is given by

$$V_W^T = [\rho_1 \rho_2 \dots \rho_K \sigma_1 \sigma_2 \dots \sigma_L] \quad (2-4)$$

Step 5: Feature vectors for all  $K \times L$  subimages of the search area of size  $M \times N$  are computed. Let  $V_{i,j}$  be the feature vector of the subimage  $S_{i,j}$ .

Step 6: Finally, an error surface of size  $(M-K+1) \times (N-L+1)$  is computed.

The elements of the error surface,  $E$ , are defined to be

$$E(i,j) = ||V_W - V_{i,j}||^2, \quad (2-5)$$

for  $1 \leq i \leq M-K+1$  and  $1 \leq j \leq N-L+1$ .

$E(i,j)$  is a measure of dissimilarity between  $W$  and  $S_{i,j}$  where minimum dissimilarity is expected at the registration point.

Step 7: Four minima in the error surface are found by finding the minimum value first. The second minimum is found by masking out an array of  $21 \times 21$  pixels centered about first minimum and then searching the remaining error surface for its minimum value.

Third and fourth minima are found similarly.

Using the sixteen pairs of reduced low and high resolution FLIR images, the above method was simulated for reference images of size  $32 \times 32$ . The method failed to identify the registration point for all scenes and therefore simulation results for this case are not included in this report.

Case 2: In order to obtain meaningful results from any of the image registration algorithms, it is necessary to preprocess the window and subimages of the search area such that the mean and standard deviation of their pixel values are equal. This is called intensity level normalization and is normally required when the window and the search area are obtained from different sensors with different d.c. gain and bias. Ideally, the window and each subimage of the search area are preprocessed to have zero mean and unity standard deviation. In this section, the effect of

intensity level normalization of the window and subimages of the search area is presented. The registration steps and simulation results are given below.

- Step 1: A reference image  $W$  of size  $K \times L$  is taken from the reduced high resolution image to include prominent features of the image.
- Step 2: Let  $\mu_W$  be the mean and  $\sigma_W$  be the standard deviation of pixel values of  $W$ . The intensity level normalized image of the window,  $W'$ , is computed as

$$W'(\ell, m) = \frac{W(\ell, m) - \mu_W}{\sigma_W}, \quad (2-6)$$

for  $1 \leq \ell \leq K$  and  $1 \leq m \leq L$ .

- Step 3: Since the window is normalized, it is no longer necessary to compute the normalized correlation between adjacent pixels of rows and columns of  $W'$ . The unnormalized correlation between the adjacent pixels of each row is computed as

$$\rho_\ell = \sum_{m=1}^{L-1} W'(\ell, m) W'(\ell, m+1)$$

for  $\ell = 1, 2, 3, \dots, K$ .

- Step 4: The unnormalized correlation between the adjacent pixels of each column is computed as

$$\sigma_m = \sum_{\ell=1}^{K-1} W'(\ell, m) W'(\ell+1, m) \quad (2-7)$$

for  $m = 1, 2, 3, \dots, L$ .

The remainder of the image registration procedure is the same as that of Case 1. Simulation results using reference images of size  $32 \times 32$  are

presented in Table 4. From Table 4 it is clear that the true registration point appeared as the first minimum for 9 out of 16 scenes. The registration point was within the first four minima for four additional scenes and was not within first four minima for the remaining three scenes. The performance of the registration method based on the correlation of adjacent pixels for intensity level normalized images is comparable with that of the binary correlation algorithm. From Case 1 and Case 2 it is concluded that intensity level normalization of images improves performance of the method and that it is not possible to use the registration algorithm based on correlation of adjacent pixels without normalizing the images. However, if each subimage of the search area is normalized independently, it is not possible to compute the feature vector of  $S_{i,j+1}$  or  $S_{i+1,j}$  from the feature vector of  $S_{i,j}$  as given in Reference 1. This increases the amount of computation for software implementation and makes the real-time hardware implementation more complex and less attractive. Therefore, an effort is made to develop an image preprocessing algorithm which has almost the same effect on registration accuracy as that of intensity level normalization of images, but allows computation of  $V_{i+1,j}$  or  $V_{i,j+1}$  from  $V_{i,j}$ . One such preprocessing algorithm is presented next.

Case 3: A possible reason for the unsatisfactory performance of the image registration method based on correlation of adjacent pixels using an unnormalized window and search area is given below. Let  $W$  and  $S_{i^*,j^*}$  be related as

$$W(x,m) = a S_{i^*,j^*}(x,m) + b, \quad (2-8)$$

$$\text{for } 1 \leq x \leq K \text{ and } 1 \leq m \leq L,$$

Table 4. Registration simulation results for correlation of adjacent pixels method.

Scene		Case 2	Case 3				Case 4
			K1=L1=5	K1=L1=7	K1=L1=9	K1=L1=11	
Position A	Target #1	1	2	3	3	2	1
	Target #2	1	X	2	2	2	1
	Target #3	1	X	X	X	3	4
	Target #4	4	1	1	1	X	1
Position B	Target #1	X	X	X	X	X	X
	Target #2	X	X	X	X	X	4
	Target #3	3	2	1	1	1	1
	Target #4	1	1	1	4	3	X
Position C	Target #1	2	X	4	1	1	1
	Target #2	X	X	2	1	1	1
	Target #3	1	1	X	1	1	1
	Target #4	1	1	1	1	1	1
Position D	Target #1	2	X	X	4	1	1
	Target #2	1	1	1	1	1	1
	Target #3	1	1	4	2	1	1
	Target #4	1	X	X	2	1	1

where  $a$  and  $b$  are constants. The above is due to the difference in gain and d.c. bias of the two imaging systems. From Equation (2-8) it is clear that the feature vectors of  $W$  and  $S_{i^*,j^*}$  are not equal if  $b \neq 0$ . However, if the two preprocessed images  $W'$  and  $S'_{i^*,j^*}$  are related by

$$W'(\ell, m) = a S'_{i^*,j^*}(\ell, m), \quad (2-9)$$

for  $1 \leq \ell \leq K$  and  $1 \leq m \leq L$ ,

then the feature vectors of  $W'$  and  $S'_{i^*,j^*}$  computed as described in Case 1 will be equal. A method of accomplishing the above is given in the steps below.

Step 1: The value of each pixel of the reduced high resolution FLIR image is replaced by its value minus the average pixel value of the  $K1 \times L1$  array centered about it. This process can be better understood from Figure 2-4 where  $X$  is the reduced high resolution FLIR image.  $X(i,j)$  is the original value of pixel  $(i,j)$  and the new value  $X'(i,j)$  is computed as

$$X'(i,j) = X(i,j) - \mu_{i,j}, \quad (2-10)$$

$$\text{where } \mu_{i,j} = \frac{1}{K1 \times L1} \left[ \sum_{k = -\frac{K1-1}{2}}^{\frac{K1-1}{2}} \sum_{\ell = -\frac{L1-1}{2}}^{\frac{L1-1}{2}} X(i+k, j+\ell) \right]$$

and where  $K1$  and  $L1$  are odd numbers.

New values for all other pixels are computed similarly. If the input image is an  $M \times N$  array, the preprocessed image will be an



$(M-Kl+1) \times (N-Ll+1)$  array of valid pixels. Reference image  $W$  of size  $K \times L$  is taken from the above  $(M-Kl+1) \times (N-Ll+1)$  array of valid pixels.

Step 2: The reduced low resolution FLIR image is preprocessed as described in Step 1 and is used as the search area with the registration procedure the same as that described in Case 1. Simulation results for a reference image of size  $32 \times 32$ , and for various values of  $Kl$  and  $Ll$  are given in Table 4. For  $Kl = Ll = 11$ , the true registration point appeared as the first minimum in the error surface for 9 out of 16 scenes simulated. Registration point was within the first four minima for four additional scenes and was not within the first four minima for the remaining three scenes. A summary of the results in Table 4 can be found in Table 6. The performance of the method for  $Kl = Ll = 11$  is as good as that of binary correlation. Since each subimage of the search area is not preprocessed separately and since intensity level normalization is no longer required after preprocessing, the feature vectors  $V_{i+1,j}$  and  $V_{i,j+1}$  can be computed from  $V_{i,j}$  with very few arithmetic operations as described in Reference 1.

Case 4: The method of computing feature vectors is the same as that of Case 3 ( $Kl = Ll = 11$ ). However, a different criterion is used for matching the feature vectors. Instead of computing  $||V_W - V_{i,j}||^2$  as a measure of dissimilarity between  $W$  and  $S_{i,j}$ ,

$$T(i,j) = \frac{V_W^T V_{i,j}}{||V_W|| \cdot ||V_{i,j}||} \quad (2-11)$$

is computed as a measure of similarity where maximum similarity is expected at the registration point. The procedure of finding the first four maxima in the correlation surface is the same as the one described earlier. The simulation results for the sixteen pairs of scenes simulated are given in Table 4. The true registration point appeared as the first maximum in the correlation surface for 12 out of 16 scenes. Registration point was within the first four maxima for two additional scenes and was not within the first four maxima for the two remaining scenes.

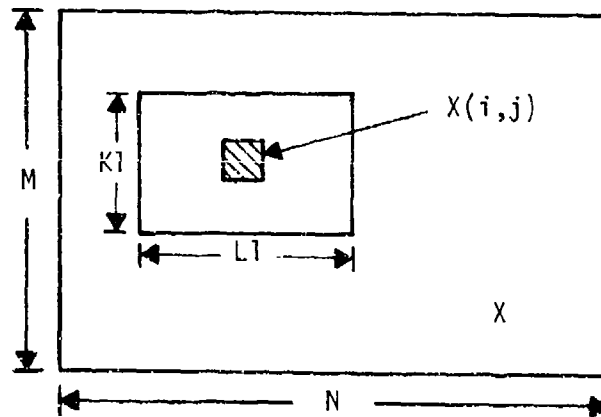


Figure 2-4. Layout for preprocessing the reduced high resolution image.

Case 5: In Case 4, each reference was chosen to include prominent features of its corresponding preprocessed reduced high resolution image. In otherwords, the reference image was not necessarily taken from the center of the field of view of the reduced high resolution image. In Case 5, the simulation is repeated by choosing the reference images from the center of field of view. Simulation results for reference images of size 32 x 32 and

32 x 64 are given in Table 5. For reference of size 32 x 32, the registration point appeared as the first maximum for only 3 out of 16 scenes. Registration point appeared as second, third or fourth maximum for two additional scenes and was not within the four maxima for the remaining eleven scenes. Even reference images of size 32 x 64 gave similar results and the performance was not improved.

#### D. Image Registration Based on Correlation of Adjacent Rows and Columns

This method is also based on correlation of adjacent pixels. Instead of computing a  $(K+L)$ -dimensional feature vector for each  $K \times L$  array based on correlation of adjacent pixels in each row and column, a  $(K+L-2)$ -dimensional feature vector is computed based on correlation of adjacent rows and adjacent columns as described in the following steps.

Step 1: The reduced high and low resolution images are preprocessed using an algorithm which replaces the value of each pixel by its value minus the average pixel value of the  $11 \times 11$  array centered about the pixel.

Step 2: The reference image  $W$  of size  $K \times L$  is chosen from the center of the field of view of the preprocessed high resolution image. The correlation between row  $i$  and row  $i+1$  is given by

$$CR(i) = \frac{\sum_{j=1}^L W(i,j) W(i+1,j)}{\left[ \sum_{j=1}^L W^2(i,j) \right]^{1/2} \left[ \sum_{j=1}^L W^2(i+1,j) \right]^{1/2}}, \quad (2-12)$$

for  $i = 1, 2, 3, \dots, K-1$ .

Table 5. Image registration using correlation of adjacent pixels.

Scene	Correlation of adjacent pixels in each row and each column		Correlation of adjacent rows and columns	
	32x32 Ref.	32x64 Ref.	32x32 Ref.	32x64 Ref.
Position A	Target #1	X -	3 -	
	Target #2	X X	X 1	
	Target #3	X X	X 2	
	Target #4	X X	1 1	
Position B	Target #1	X X	4 X	
	Target #2	X X	X X	
	Target #3	X X	X 1	
	Target #4	1 2	X 2	
Position C	Target #1	4 X	1 1	
	Target #2	X 4	X X	
	Target #3	2 1	1 1	
	Target #4	1 1	1 2	
Position D	Target #1	X X	X X	
	Target #2	X X	X X	
	Target #3	X X	X 4	
	Target #4	X X	X X	

Step 3: Similarly, let  $CC(j)$  be the normalized correlation between column  $j$  and column  $j+1$  as given by

$$CC(j) = \frac{\sum_{i=1}^K W(i,j) W(i,j+1)}{\left[ \sum_{i=1}^K W^2(i,j) \right]^{1/2} \left[ \sum_{i=1}^L W^2(i,j+1) \right]^{1/2}}, \quad (2-13)$$

for  $j = 1, 2, 3, \dots, L-1$ .

Step 4: The  $(K+L-2)$ -dimensional feature vector,  $V_W$ , of the digital image  $W$  is given by

$$V_W^T = [CR(1) \ CR(2) \ \dots \ CR(K-1) \ CC(1) \ CC(2) \ \dots \ CC(L-1)] \quad (2-14)$$

Step 5: Feature vectors are computed for all subimages of the search area and matched with those of the reference.

A total of sixteen pairs of scenes are simulated using the above method for reference images of  $32 \times 32$  with results presented in Table 5. The registration point appeared as the first maximum for four scenes, and was within the first four maxima for two additional scenes. Registration point was not within first four maxima for ten remaining scenes. When the reference image size was increased to  $32 \times 64$ , the registration point appeared as the first maximum in the correlation surface for five scenes, was within the first four maxima for four additional scenes, and was not within the first four maxima for the remaining six scenes. From Table 5, it is clear that the performance of the method based on the correlation of adjacent rows and columns is better than the method based on correlation of adjacent pixels in each row and each column. The above method is

expected to do better if the scenes consist of prominent horizontal and vertical edges. A summary of the results in Tables 4 and 5 is presented in Table 6.

#### E. Image Registration Using Moments Method

A method of accomplishing digital image registration based on moments of the reference  $W$  and subimages of the search area  $S$  is given in Reference 1. For multiple image registration, it has been shown that the moments method requires fewer arithmetic operations than the standard correlation algorithm. However, not much is known about the accuracy and reliability of the method which may be very sensitive to noise. The reliability of the method is investigated through simulation using the same 16 typical military type scenes used in earlier simulations. An attempt is made to accomplish digital image registration without intensity level normalization of the window and the search area as described in the steps below.

- Step 1: A reference image of size  $K \times L$  ( $32 \times 32$ ) is taken from the reduced high resolution FLIR image to include the prominent features of the image.
- Step 2: All moments of  $W$  of order three and less are computed using the relation,

$$m_{pq}^W = \sum_{\ell=1}^K \sum_{m=1}^L \ell^p m^q W(\ell, m), \quad (2-15)$$

for  $p, q = 0, 1, 2$  and  $3$  such that  $p + q \leq 3$ .

Table 6. Summary of results in Table 4 and Table 5.

Registration Method	Registration Point is 1st Maximum	Registration Point is 2nd, 3rd or 4th Maximum	Registration Point not in first four Maxima
Case 2	9	4	3
Registration method based on the corre- lation of adjacent pixels in each row and each column.	K1=L1=5	2	2
	K1=L1=7	5	6
	K1=L1=9	6	3
	K1=L1=11	4	3
Case 4	12	2	2
Case 5	32x32 ref.	2	12
	32x64 ref.	2	11
Reference	4	2	10
Registration method based on the corre- lation of adjacent rows and columns.	32x32		
	32x64	4	6

Step 3: All moments of order 3 and less are computed for each subimage of the search area. Let  $\{m_{pq}^{ij}\}$  be the moment sequence of the subimage  $S_{i,j}$ .

Step 4: The moments of the subimages of  $S$  are matched with those of the reference image. The degree of dissimilarity between  $W$  and  $S_{i,j}$  is computed as

$$E(i,j) = \sum_{p,q} (m_{pq}^W - m_{pq}^{ij})^2, \quad (2-16)$$

for  $1 \leq i \leq M-K+1$ ,  $1 \leq j \leq N-L+1$ ,

where  $M$  and  $N$  are the number of rows and columns in the search area. Minimum error is expected at the registration point.

A total of sixteen pairs of scenes were simulated and the method failed for each pair. Even though the reference images were chosen to include the prominent features of the reduced high resolution images, the true registration points did not appear within the first four minima in the error surface. The simulation was repeated using intensity level normalized windows and search areas without any success. Therefore, it is concluded that moments method is not suitable for image registration for the kind of images considered in this report, at least not without additional preprocessing such as smoothing which was not tried.

#### F. Image Registration Using Intraset and Interset Distances

A method of accomplishing multiple image registration using characteristic features, namely the intraset and interset distances is discussed in detail in [1]. Since features are first extracted and matching is performed on the feature set only, this method is computationally more



efficient than the correlation methods for multiple image registration. However, the accuracy and reliability of feature matching algorithms depend very much on the efficient selection of features. The registration accuracy and reliability of this method for bilevel and trilevel images are investigated through simulation using typical military scenes described in a previous section. The registration technique used and the simulation results are presented below.

- Step 1: A reference image  $W$  of size  $K \times L$  is selected from the reduced high resolution FLIR image.
- Step 2: The mean  $\mu_W$  and the standard deviation  $\sigma_W$  of the pixel values of  $W$  are computed. The window  $W$  is segmented into three sets  $S_0$ ,  $S_1$ , and  $S_2$ .

$$W(\ell, m) \in \begin{cases} S_0, & \text{if } W(\ell, m) < \mu_W - S\sigma_W \\ S_1, & \text{if } W(\ell, m) > \mu_W + S\sigma_W \\ S_2, & \text{otherwise} \end{cases} \quad (2-17)$$

for  $1 \leq \ell \leq K$  and  $1 \leq m \leq L$ .

Where  $S$  is a scale factor greater than zero. The reference image  $W$  can be segmented into only two sets  $S_0$  and  $S_1$  by setting  $S$  equal to zero.

- Step 3: The intraset distances of sets  $S_0$ ,  $S_1$ , and  $S_2$  and the intersets distances between  $S_0$  and  $S_1$ ,  $S_0$  and  $S_2$ , and  $S_1$  and  $S_2$  are computed [1]. Let  $D_x(S_i)$  and  $D_y(S_i)$ ,  $i = 0, 1, 2$  be the  $x$  and  $y$  components of the intraset distances, and  $D_x(S_i, S_j)$  and  $D_y(S_i, S_j)$ ,  $i \neq j$  for  $i, j = 0, 1, 2$  be  $x$  and  $y$  components of

the interset distances. The 12 components of intraset and inter-set distances form a 12-dimensional feature vector  $V_W$  of the reference image  $W$ .

Step 4: A similar feature vector  $V_{i,j}$  for each subimage  $S_{i,j}$  of size  $K \times L$  of the reduced low resolution FLIR image (search area) is computed according to steps 2 and 3.

Step 5: The error surface  $E$  is generated by computing the Euclidean distance between  $V_W$  and  $V_{i,j}$  for all subimages of the search area  $S$ .

Step 6: The registration point is expected to have minimum error.

Reduced low resolution FLIR images of size  $406 \times 313$  and reduced high resolution FLIR images of size  $98 \times 78$  described earlier in the report are used in the simulation. For each pair of scenes (search area and window) the following three cases are considered.

Case 1: Reference arrays of size  $16 \times 16$  are chosen to include the prominent features in the reduced high resolution FLIR images. The windows and subimages of the search area are segmented into two levels by setting  $S$  equal to zero in Equation (2-17). The feature vector, therefore, consists of one interset and two intraset distances.

Case 2: Reference arrays of size  $24 \times 24$  are chosen to include the prominent features in the reduced high resolution FLIR images. The segmentation of images is the same as Case 1.

Case 3: Reference arrays of size  $24 \times 24$  are chosen to include the prominent features in the reduced high resolution FLIR images and

are segmented into three levels  $S_0$ ,  $S_1$ , and  $S_2$  by setting  $S$  equal to 0.1.

Case 4: 24 x 24 reference arrays and subimages of the search area are segmented into three levels  $S_0$ ,  $S_1$ , and  $S_2$  by setting  $S$  equal to 0.1. However, the set  $S_2$  is not considered in the registration process. In other words, the feature vectors consist only of the two intraset distances of sets  $S_0$  and  $S_1$ , and an interset distance between  $S_0$  and  $S_1$ .

The simulation results for each of the above cases is given in detail in the monthly reports of December 1980 and January 1981 and are summarized in Tables 7 and 8. In Table 7, the appearance of the registration point as 1st, 2nd, 3rd and 4th minima in the error surface is indicated by 1, 2, 3, and 4, respectively. "X" indicates that the registration point did not appear within first four minima. A few simulations were also run for reference arrays of size 32 x 32 and 32 x 64 with extremely poor results. From this result and the results summarized in Tables 7 and 8 it is concluded that this method is not accurate for the type of imagery considered and will not be discussed further.

#### G. A New Feature Matching Image Registration Technique

A new method of accomplishing image registration using features based on adjacent pixel differences along the rows and columns of a digital image is presented. In the feature matching technique, a digital image is represented by  $R$  features or an  $R$ -dimensional feature vector, and the decision-making process is based on a similarity measure which,

Table 7. Simulation results of intraset and intersset distance methods.

Scene		Case 1	Case 2	Case 3	Case 4
Position A	Target #1	1	X	2	4
	Target #2	1	1	2	1
	Target #3	X	X	X	X
	Target #4	X	X	X	X
Position B	Target #1	X	X	X	X
	Target #2	4	4	X	2
	Target #3	X	3	1	4
	Target #4	X	X	X	X
Position C	Target #1	X	X	X	X
	Target #2	1	X	X	2
	Target #3	1	X	4	1
	Target #4	1	1	2	1
Position D	Target #1	1	X	4	X
	Target #2	X	2	3	3
	Target #3	X	X	2	2
	Target #4	X	X	X	X

Table 8. Summary of intraset and interset simulation results presented in Table 7.

	Registration Point is the first minimum	Registration Point is 2nd, 3rd, or 4th Minimum	Registration Point not in first four minima
Case 1	6	1	9
Case 2	2	3	11
Case 3	1	7	8
Case 4	3	6	7

in turn, is expressed in terms of a distance measure (Euclidean distance) or the correlation of feature vectors (cosine of the angle between the two vectors).

Let  $W$  be a digital image (window) of size  $K \times L$  whose pixels can assume one of  $G$  possible levels on the gray scale

$$0 \leq W(i,j) \leq G-1$$

for  $1 \leq i \leq K$  and  $1 \leq j \leq L$ .

The feature elements of each row and column of  $W$  are computed using equations (2-18) and (2-19), respectively.

$$R_k = \frac{\sum_{j=1}^{L-1} [W(k,j) - W(k,j+1)]^2}{\sum_{j=1}^L W(k,j)^2}, \text{ for } k = 1, 2, \dots, K \quad (2-18)$$

and

$$C_\ell = \frac{\sum_{i=1}^{K-1} [W(i,\ell) - W(i+1,\ell)]^2}{\sum_{i=1}^K W(i,\ell)^2}, \text{ for } \ell = 1, 2, \dots, L \quad (2-19)$$

The feature element  $R_k$  ( $C_\ell$ ) is always greater than or equal to zero, assuming zero value only when all the pixels in the  $k$ th row ( $\ell$ th column) of  $W$  are at the same level on the gray scale.

A computationally more efficient way of extracting row and column features of a digital image of size  $K \times L$  is to use Equations (2-20) and (2-21) instead of (2-18) and (2-19), respectively.

$$R_k = \frac{\sum_{j=1}^{L-1} |W(k,j) - W(k,j+1)|}{\sum_{j=1}^L |W(k,j)|}, \text{ for } k = 1, 2, \dots, K \quad (2-20)$$

and

$$C_\ell = \frac{\sum_{i=1}^{K-1} |W(i,\ell) - W(i+1,\ell)|}{\sum_{i=1}^K |W(i,\ell)|}, \text{ for } \ell = 1, 2, \dots, L \quad (2-21)$$

The  $(K+L)$ -dimensional feature vector,  $V_W$  of the digital image  $W$  is now formed as given by

$$V_W^T = [R_1 \ R_2 \ \dots \ R_K \ C_1 \ C_2 \ \dots \ C_L] \quad (2-22)$$

This feature vector is correlated with the feature vectors of the sub-images of size  $K \times L$  of the search area to determine the registration point.

#### 1. Derivation of the Computational requirements for Multiple image registration

The number of additions and multiplications required to implement the new feature matching algorithm for a search area of size  $M \times N$  and  $n$  reference images (windows) of size  $K \times L$  are derived in the following steps:

1. Computation of  $R_k$  as given by Equation (2-20) requires  $(3L-4)$  additions (subtractions) and one multiplication (division). Similarly,  $(3K-4)$  additions and one multiplication are needed to compute  $C_\ell$ . Since there are  $K$  rows and  $L$  columns, the total number of additions and

multiplications required to compute the feature vector  $V_W$  of a digital image  $W$  of size  $K \times L$  are given by

$$\begin{aligned} \text{Total number of additions} &= K(3L-4) + L(3K-4) \\ &= 6KL - 4(K+L) \end{aligned}$$

$$\text{Total number of multiplications} = K+L$$

2. In order to compute the feature vector  $V_{i,j}$  of the subimage  $S_{i,j}$  also of size  $K \times L$ ,  $6KL - 4(K+L)$  additions and  $(K+L)$  multiplications are required as outlined in step 1.
3. The feature vector  $V_{i,2}$  of the subimage  $S_{i,2}$  can be computed with very few arithmetic operations once  $V_{i,1}$ , the feature vector of subimage  $S_{i,1}$ , is computed.

Now,

$$R_k(i,1) = \frac{\sum_{j=1}^{L-1} |S_{i,1}(k,j) - S_{i,1}(k,j+1)|}{\sum_{j=1}^L |S_{i,1}(k,j)|} = \frac{N_k(i,1)}{D_k(i,1)} \quad (2-23)$$

for  $k = 1, 2, \dots, K$ .

where

$$N_k(i,1) = \sum_{j=1}^{L-1} |S_{i,1}(k,j) - S_{i,1}(k,j+1)|, \text{ and}$$

$$D_k(i,1) = \sum_{j=1}^L |S_{i,1}(k,j)|$$



Then

$$\begin{aligned}
 R_k(i,2) &= \frac{\sum_{j=1}^{L-1} |S_{i,2}(k,j) - S_{i,2}(k,j+1)|}{\sum_{j=1}^L |S_{i,2}(k,j)|} \\
 &= \frac{N_k(i,1) - |S_{i,1}(k,1) - S_{i,1}(k,2)| + |S_{i,2}(k,L-1) - S_{i,2}(k,L)|}{D_k(i,1) - |S_{i,1}(k,1)| + |S_{i,2}(k,L)|}
 \end{aligned}
 \tag{2-24}$$

It is seen from Equation (2-24), that  $R_k(i,2)$  can be computed from  $R_k(i,1)$  and in general  $R_k(i,j)$  can be computed from  $R_k(i,j-1)$  by performing only 6 equivalent addition and one multiplication operations. Computation of the  $K$  row-feature elements of  $S_{i,2}$ , therefore, requires  $6K$  additions and  $K$  multiplications. Also, since

$$C_\ell(i,2) = C_{\ell+1}(i,1), \text{ for } \ell = 1, 2, \dots, L-1,$$

it is only necessary to compute  $C_L(i,2)$  which requires  $(3K-4)$  additions and one multiplication as shown in Step 1. The computation of the feature vector  $V_{i,2}$  from  $V_{i,1}$ , therefore, requires a total of  $(9K-4)$  additions and  $(K+1)$  multiplications.

4. There are  $(M-K+1)(N-L+1)$  reference points or subimages in a search area of size  $M \times N$ . The computation of the feature vector associated with the reference point  $(1,1)$  requires  $6KL - 4(K+L)$  additions and  $K+L$  multiplications (Step 1). The computation of the remaining  $(N-L)$  reference points in the first row requires a total of  $(9K-4)(N-L)$  additions and  $(K+1)(N-L)$  multiplications. The computation of the feature vectors of all the  $(M-K+1)(N-L+1)$  subimages, therefore, requires

$(M-K+1)[6KL-4(K+L) + (9K-4)(N-L)]$  additions, and  
 $(M-K+1)[(K+L) + (K+1)(N-L)]$  multiplications . (2-25)

5. The correlation surface is generated by correlating the reference feature vector with each of the  $(M-K+1)(N-L+1)$  feature vectors of the subimages using the relation

$$C_{i,j} = \frac{(V_W^T V_{i,j}) | (V_W^T V_{i,j}) |}{||V_W||^2 ||V_{i,j}||^2}, \text{ for all } i,j \quad (2-26)$$

Computation of  $C_{i,j}$  for feature vectors of dimension  $(K+L)$  requires  $(3K+3L-3)$  additions and  $(3K+3L+5)$  multiplications. The generation of the correlation surface, therefore, requires a total of

$(M-K+1)(N-L+1)(3K+3L-3)$  additions and  
 $(M-K+1)(N-L+1)(3K+3L+5)$  multiplications.

6. Assuming that there are  $n$  windows, the multiple image registration process requires Steps 1 and 5 to be performed  $n$  times. Therefore, for multiple image registration,

$$\begin{aligned} \text{Total number of additions required} &= \\ &= n[6KL-4(K+L)] \\ &+ (M-K+1)[6KL-4(K+L)+(9K-4)(N-L)] \\ &+ n(M-K+1)(N-L+1)(3K+3L-3) \end{aligned}$$

$$\begin{aligned} \text{Total number of multiplications required} &= \\ &= n(K+L) \\ &+ (M-K+1)[(K+L)+(K+1)(N-L)] \\ &+ n(M-K+1)(N-L+1)(3K+3L+5) \end{aligned} \quad (2-27)$$

for  $M = 240$ ,  $N = 256$ ,  $K = 32$ , and  $L = 32$

Total number of additions =  $8893613n + 14526336$

Total number of multiplications =  $9263989n + 1558304$

The number of additions and multiplications which are required to implement correlation, SSDA, moments method and the new feature matching method for various values of  $n$ , are shown in Figures (2-5) and (2-6), respectively. Assuming that multiplication, division, and squaring operations are equivalent and that the time required to perform one real multiplication is three times that required to perform one real addition, the total number of equivalent real additions required to implement the various methods are shown in Figure (2-7). It can be concluded from Figures (2-5) through (2-7) that, the new feature matching technique is computationally more efficient than any of the template matching or feature matching methods considered so far for multiple image registration.

## 2. Simulation Results

An effort is made to study the performance and registration accuracy of the new feature matching method through simulation. Reduced low resolution FLIR images (search area) of size  $406 \times 313$  and reduced high resolution FLIR images of size  $98 \times 78$  are used for simulation. The registration technique and the simulation results are presented below for all the cases considered.

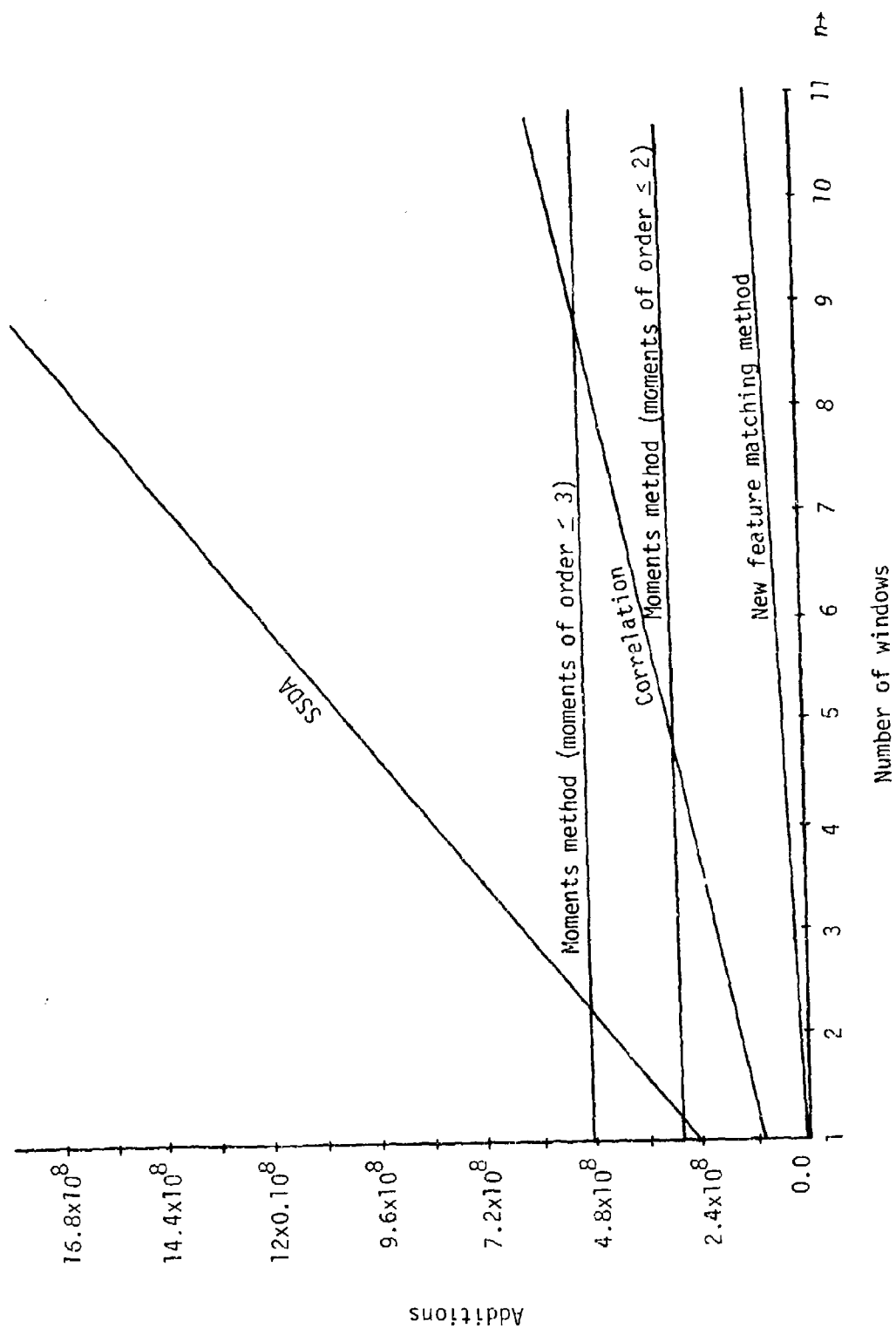


Figure 2-5. Number of additions to register  $n$  windows.

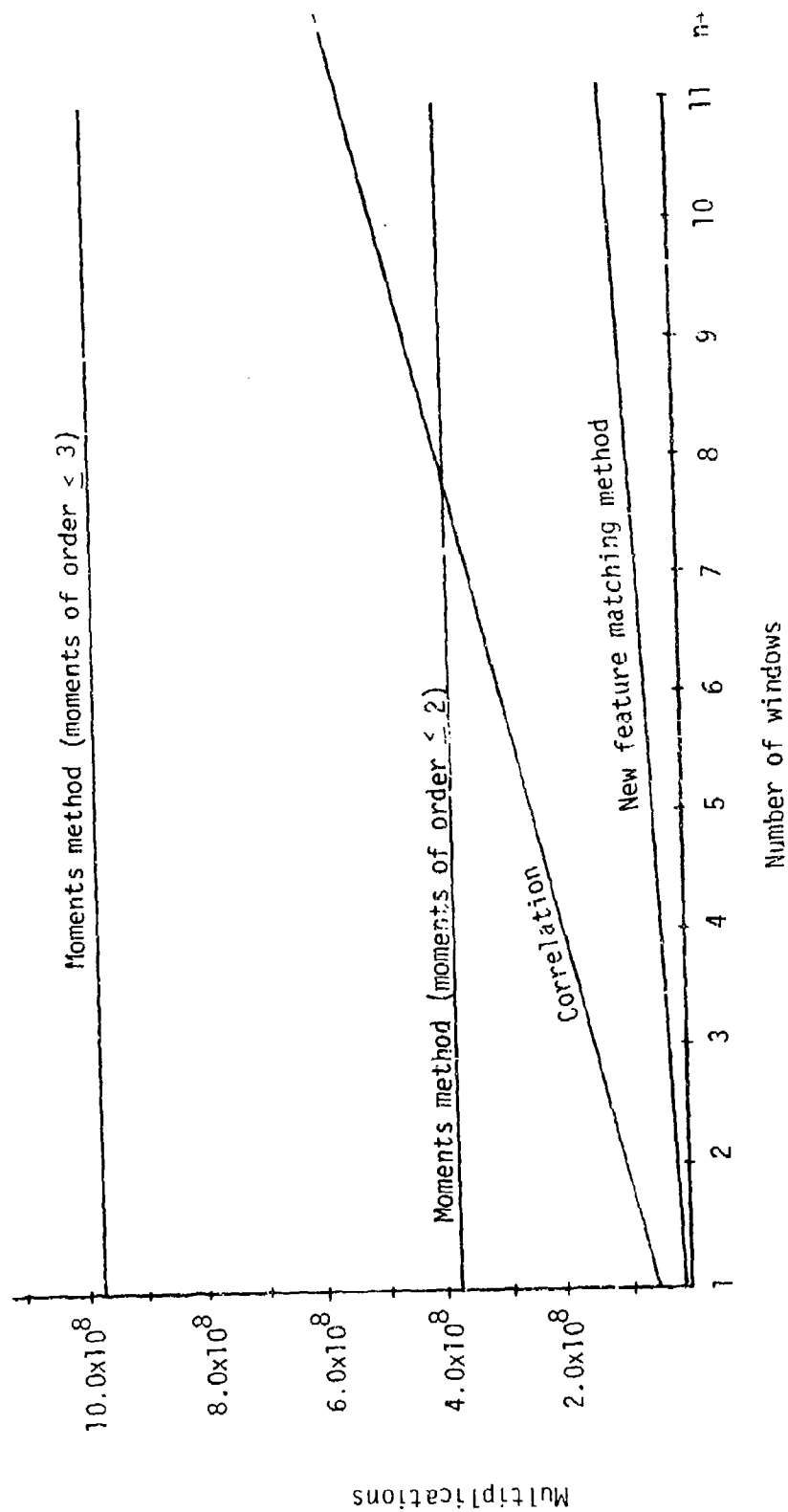


Figure 2-6. Number of multiplications to register  $n$  windows.

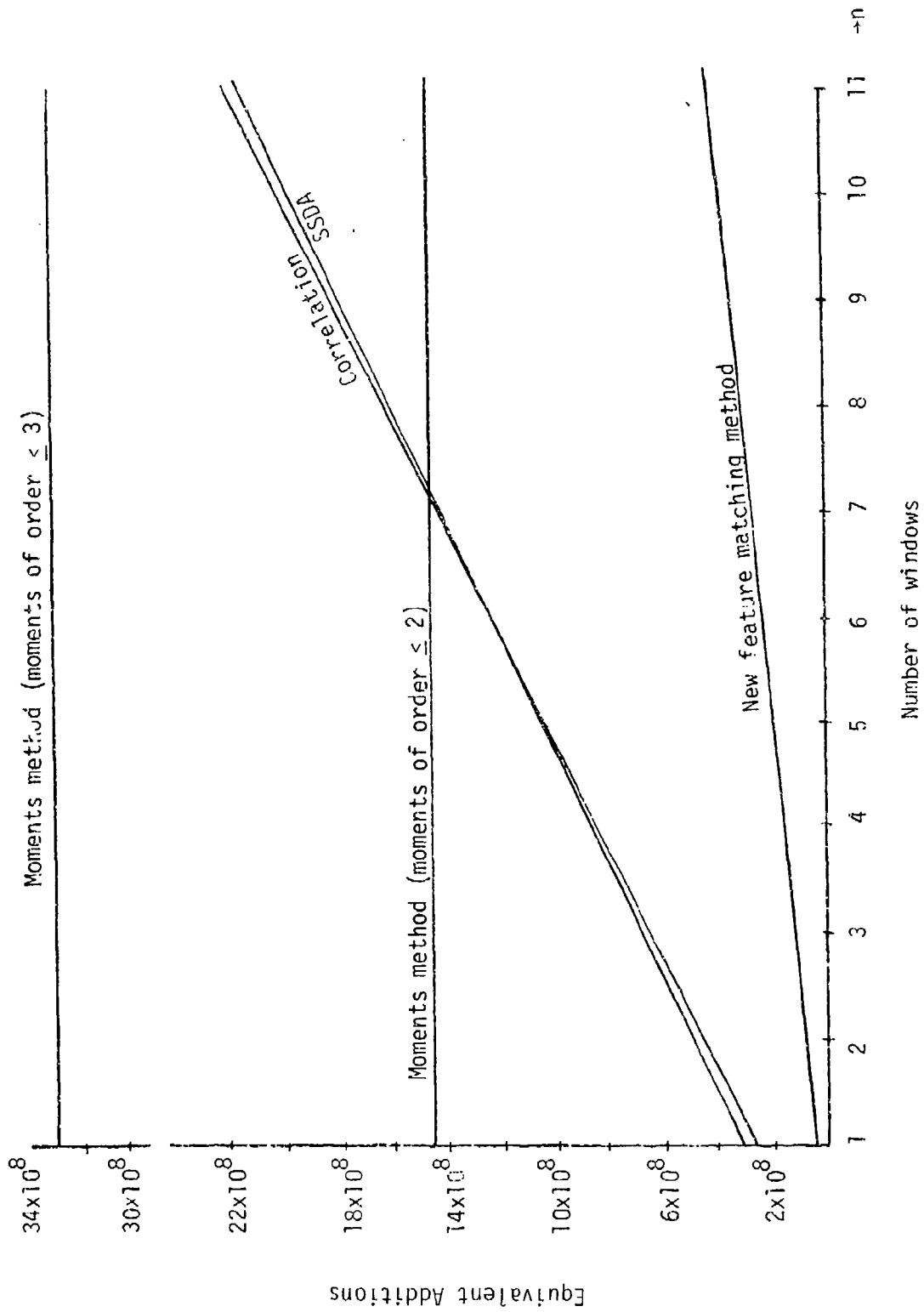


Figure 2-7. Number of equivalent additions to register  $n$  windows.

Case 1: Reduced low and high resolution FLIR images are used for the simulation purpose without intensity level normalization or any other preprocessing. The registration technique is described below.

Step 1: A reference image  $W$  of size  $K \times L$  is selected from the reduced high resolution FLIR image.

Step 2: The feature elements of each row and column of  $W$  are computed using Equations (2-18) and (2-19), respectively.

Step 3: The  $(K+L)$ -dimensional feature vector,  $V_W$ , of the reference image  $W$  is now formed as given by

$$V_W^T = [R_1, R_2, \dots, R_K, C_1, C_2, \dots, C_L]$$

Step 4: A subarray of size  $120 \times 240$  which includes the target area of interest is extracted from the  $406 \times 313$  reduced low resolution FLIR image and is used as the search area.

Step 5: The  $(K+L)$ -dimensional feature vectors,  $V_{i,j}$ , for each subimage  $S_{i,j}$  of size  $K \times L$  is computed as in Steps 2 and 3 and the cosine of the angle between the two feature vectors  $V_W$  and  $V_{i,j}$  is computed using the relation

$$C_{i,j} = \frac{V_W^T V_{i,j}}{\|V_W\| \|V_{i,j}\|}, \text{ for all } i,j$$

Step 6:  $C_{i,j}$  is expected to have a maximum at the registration point. Simulations were run for a reference array of size  $32 \times 32$ , choosing the reference from the center of the field of view of the reduced high resolution FLIR image.

Case 2: As in Case 1 with 32 x 64 reference images.

Case 3: Case 3 is similar to Case 1 except that the 64-dimensional feature vector,  $V_W$ , of the reference image  $W$  of size 32 x 32 is formed by computing the row and column feature elements using Equations (2-20) and (2-21) instead of Equations (2-18) and (2-19), respectively. The feature vectors  $V_{i,j}$  for each sub-image  $S_{i,j}$  of size 32 x 32 are also computed using Equations (2-20) through (2-22).

Case 4: As in Case 3 with 32 x 64 reference images.

Case 5: Since the two images, the window  $W$  and its matching subimage  $S_{i,j}$  of the search area are obtained from different sensors, the corresponding pixels in the two images may not have the same pixel value due to the difference in gain and dc bias of the two imaging systems. In such a case it is reasonable to assume that  $S_{i,j}$  and  $W$  are related by the following equation

$$W(\ell, m) = a S_{i,j}(\ell, m) + b, \quad 1 \leq \ell \leq K \text{ and } 1 \leq m \leq L$$

where  $a$  (gain) and  $b$  (dc bias) are constants. Since the numerators of Equations (2-20) and (2-21) are computed by taking the difference between adjacent pixels, they are independent of the constant  $b$ . Since the denominators are functions of the constant  $b$ , the matching of feature vectors of  $W$  and  $S_{i,j}$  may lead to false registration. This false registration can be avoided by preprocessing the two images so as to have zero means or equal means. The constant multiplication factor " $a$ " does not result in false registration because the decision making process is



based on a similarity measure which is expressed as the correlation of feature vectors (see the Equation in Step 5).

The registration process is identical to that of Case 3, except that both the high resolution and low resolution FLIR images are preprocessed using a moving area-average filter before the extraction of the 64-dimensional feature vector. Preprocessing of the two images is done using an algorithm which replaces the value of each pixel by its value minus the average pixel value of the  $11 \times 11$  array centered about the pixel. The two preprocessed images will then have zero or nearly equal means. Simulation results indicated a significant improvement in the performance when preprocessed images are used.

Case 6: Reference image of size  $32 \times 64$  chosen from the center of the field of view of the reduced high resolution FLIR image is used for the simulation purpose. The registration technique is similar to Case 5. Both the high resolution and low resolution FLIR images are preprocessed using a moving area-average filter described in Case 5 before the extraction of the 96-dimensional feature vector.

Case 7: The registration technique is similar to Case 4 except that the feature vector is computed in a slightly different way. The  $32 \times 64$  array is processed as two separate blocks of size  $32 \times 32$  and the registration process is described below.

Step 1: The reference image  $W$  of size  $32 \times 64$  is divided into two sub-images  $W_1$  and  $W_2$  of size  $32 \times 32$  each.

Step 2: The feature elements of each row and column of  $W_1$  and  $W_2$  are computed using Equations (2-28) and (2-29), respectively.

$$R_{k,\ell} = \frac{\sum_{j=1}^{32-1} |[W_{\ell}(k,j) - W_{\ell}(k,j+1)]|}{\sum_{j=1}^{32} |W_{\ell}(k,j)|} \quad (2-28)$$

and

$$C_{k,\ell} = \frac{\sum_{i=1}^{32-1} |[W_{\ell}(i,k) - W_{\ell}(i+1,k)]|}{\sum_{i=1}^{32} |W_{\ell}(i,k)|} \quad (2-29)$$

for  $k = 1, 2, 3, \dots, 32$ , and  $\ell = 1, 2$ .

Step 3: The 128-dimensional feature vector,  $V_W$ , of the reference image  $W$  is now formed as given by

$$V_W^T = [R_{1,1} \ R_{2,1} \ \dots \ R_{32,1} \ R_{1,2} \ R_{2,2} \ \dots \ R_{32,2} \ C_{1,1} \ C_{2,1} \ \dots \ C_{32,1} \\ C_{1,2} \ C_{2,2} \ \dots \ C_{32,2}]$$

Step 4: A 128-dimensional feature vector,  $V_{i,j}$ , for each subimage of size  $32 \times 64$  from the search area is computed according to Steps 2 and 3.

Step 5: The two feature vectors  $V_W$  and  $V_{i,j}$  are correlated to find the registration point.

The simulation results for each of the above cases is given in detail in the monthly reports of May through July 1981 and are summarized in Tables 9 and 10. The simulation results indicate that the new feature matching technique performs better than the binary correlation method and at the same time requires less computation since the feature vectors

Table 9. Simulation results of new feature matching image registration technique.

Scenes		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
Position A	Target #1	X	-	X	-	X	-	-
	Target #2	X	1	X	1	X	1	1
	Target #3	3	2	2	1	2	1	1
	Target #4	4	X	X	X	X	X	4
Position B	Target #1	X	X	X	X	X	4	3
	Target #2	1	2	4	1	1	1	1
	Target #3	1	3	1	2	1	3	1
	Target #4	1	1	1	2	1	2	X
Position C	Target #1	3	3	1	1	1	4	1
	Target #2	4	1	2	2	1	1	1
	Target #3	1	2	1	1	1	1	1
	Target #4	1	2	1	1	1	1	1
Position D	Target #1	X	X	3	X	X	X	4
	Target #2	X	X	X	X	X	X	2
	Target #3	X	X	X	X	3	X	4
	Target #4	X	X	X	X	X	X	X

Table 10. Summary of results in Table 9.

Registration Technique	Registration Point is First Maximum	Registration Point is 2nd, 3rd, or 4th Maximum	Registration Point not in First Four Maxima
Case 1 - Feature matching (32x32 reference image)	5	4	7
Case 2 - Feature matching (32x64 reference image)	3	6	6
Case 3 - Modified feature matching (32x32 reference image)	5	4	7
Case 4 - Modified feature matching (32x64 reference image)	6	3	6
Case 5 - Modified feature matching with moving average filter (32x32 reference image)	7	2	7
Case 6 - Modified feature matching with moving average filter (32x64 reference image)	6	4	5
Case 7 - Modified feature matching with moving average filter (two 32x32 reference image blocks)	8	5	2

are generated recursively. The performance is still improved with the modified features where the absolute value of the difference between adjacent pixels is taken instead of the squared value resulting in further reduction in computation. Preprocessing of the images using a moving area-average filter before the extraction of the features resulted in a significant improvement in the registration accuracy of the modified feature matching technique. Further improvement in the registration accuracy when a 32 x 64 image array is processed as two separate blocks of size 32 x 32 is evident from the results summarized in Table 9. Based on the above facts it can be concluded that for the type of military scenes considered the new modified-feature matching technique with preprocessed images performs better than all the methods considered so far, where the reference is taken from the center of the FOV, and at the same time requires less computation time, thus making the method more attractive for real-time implementation.

#### H. Image Registration Using Haar Coefficients

Orthogonal transforms, such as the Haar transform (HT), convert a digital image in the "spatial-domain" into a "transform-domain" image that contains the same information in a nonspatial representation. The Haar transform is derived from the Haar matrix [6] whose elements are either 1, -1, or 0 multiplied by powers of  $\sqrt{2}$ . An 8 x 8 Haar matrix is shown in Figure 2-8.

$$H = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Figure 2-8. An 8 x 8 Haar matrix.

The two dimensional Haar transform  $[F]$  of a digital image  $[f]$  is defined by

$$[F] = H[f] H^T \quad (2-30)$$

where  $[F]$ ,  $[f]$ , and  $H$  are all of the same order.

The Haar transform can be visualized as a course-to-fine sampling process of the input data sequence, where the first transform-coefficient represents the mean value of the components of the sequence, the second represents the average difference of the first  $N/2$  components and the second  $N/2$  components, and so on, with the last  $N/2$  transform-coefficients measuring the adjacent differences of the data elements taken two at a time. The Haar transform coefficients may be called the sequency (spatial frequency) components and are analogous to the Fourier transform coefficients which are called the frequency components. The sequency of any row (or column) of a Haar matrix is the number of sign changes along that row (or column) [10]. One of the several properties of Haar transform is its ability to store a large percentage of the image information in a few

coefficients, so that a feature vector with only a small number of elements is required for retention of significant image information. The criterion for the selection of features for matching purposes may be one of the following:

1. Select all transform coefficients whose absolute values are greater than a predetermined threshold.
2. Select those transform coefficients which have the largest variation or variance across image classes.
3. Select a predetermined number of highest transform coefficients.
4. Choose the low sequency components assuming that a major portion of the high sequency components represent noise.

Computation of the two dimensional Haar transform of a digital image can be best illustrated by using the two dimensional basis plane as shown in Figure 2-9 for an 8 x 8 transform. Each of the transform coefficients may then be considered proportional to the correlation of the image and the corresponding basis plane. For example, the (3, 4) coefficient in the transform-domain image is computed as follows. Every element of the 8 x 8 spatial-domain image of Figure 2-10 is multiplied by the corresponding element in the (3, 4) basis plane shown in Figure 2-11 and the products are added together. The resulting product is then multiplied by the proper weighting factor which is 2 for the (3, 4) coefficient as indicated in Figure 2-12. The (3, 4) coefficient of the transform domain image is therefore given by

$$F(3, 4) = 2(f_{15} + f_{16} - f_{17} - f_{18} + f_{25} + f_{26} - f_{27} - f_{28} \\ - f_{35} - f_{36} + f_{37} + f_{38} - f_{45} - f_{46} + f_{47} + f_{48}) .$$

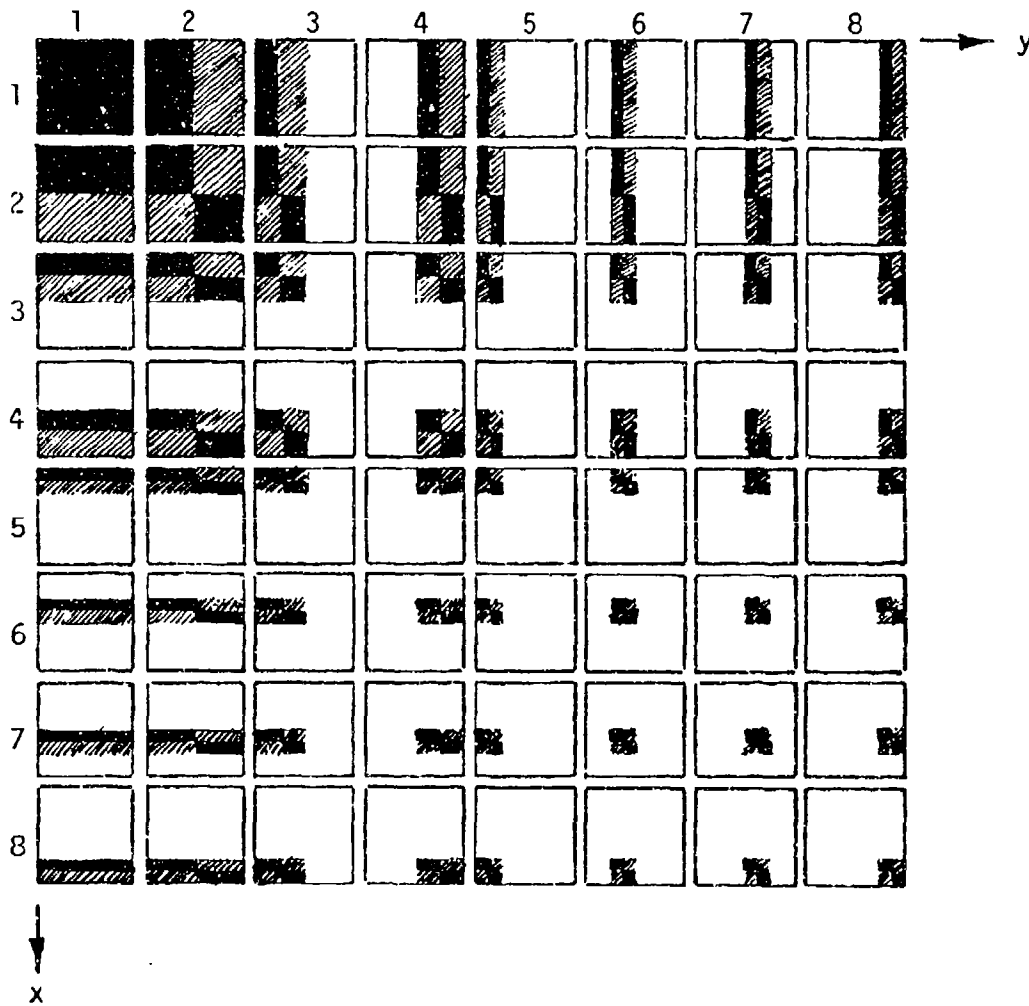
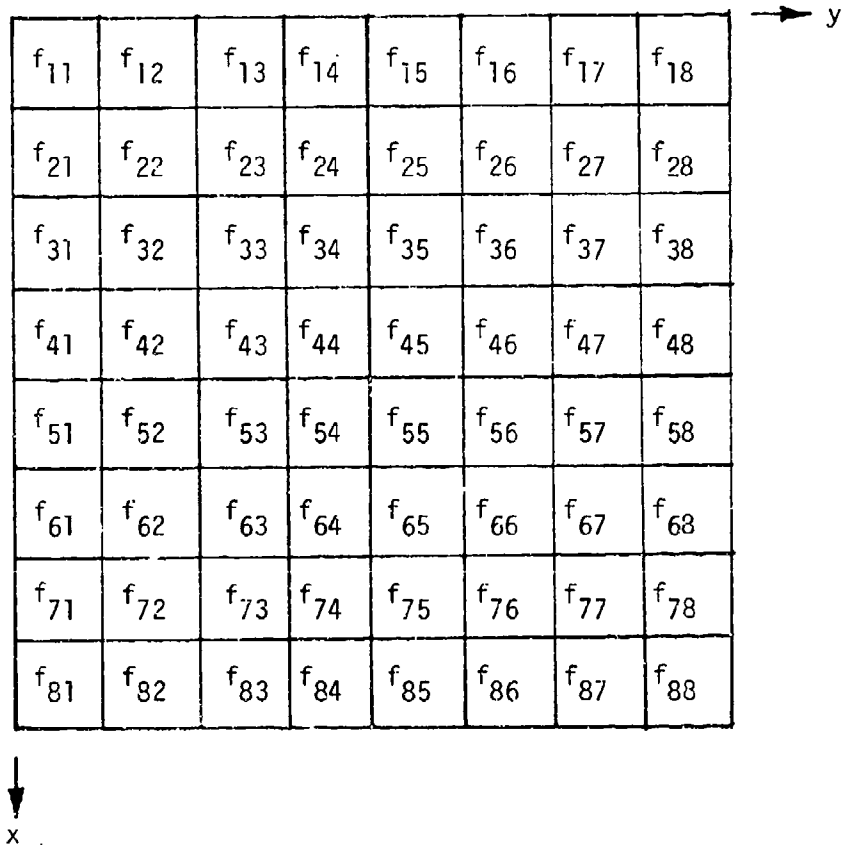


Figure 2-9. Basis Plane for an 8x8 Haar transform.  
Black=+1; crosshatched=-1; white=0.

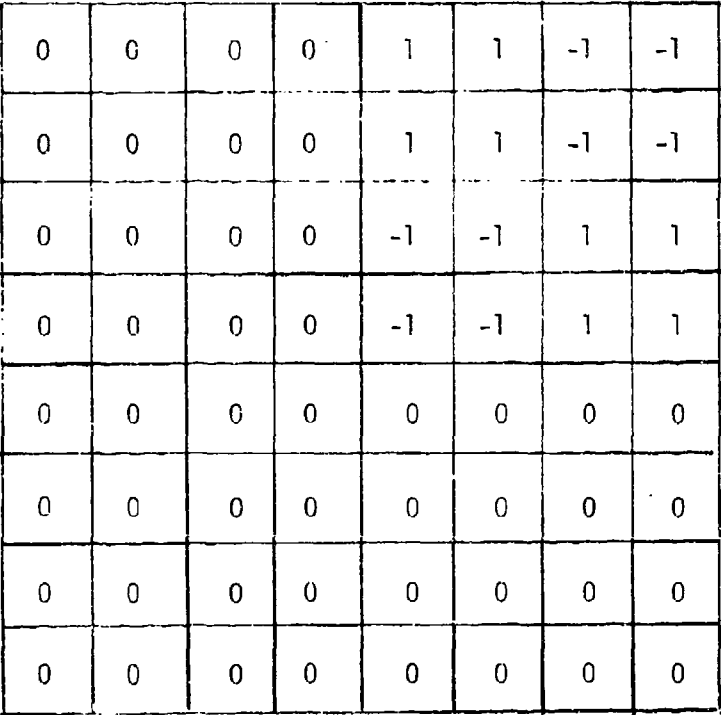




An 8x8 grid representing a spatial domain digital image. The grid is labeled with  $f_{ij}$  where  $i$  is the row index (1 to 8) and  $j$  is the column index (1 to 8). The horizontal axis is labeled  $y$  with an arrow pointing to the right, and the vertical axis is labeled  $x$  with an arrow pointing downwards.

$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$	$f_{18}$
$f_{21}$	$f_{22}$	$f_{23}$	$f_{24}$	$f_{25}$	$f_{26}$	$f_{27}$	$f_{28}$
$f_{31}$	$f_{32}$	$f_{33}$	$f_{34}$	$f_{35}$	$f_{36}$	$f_{37}$	$f_{38}$
$f_{41}$	$f_{42}$	$f_{43}$	$f_{44}$	$f_{45}$	$f_{46}$	$f_{47}$	$f_{48}$
$f_{51}$	$f_{52}$	$f_{53}$	$f_{54}$	$f_{55}$	$f_{56}$	$f_{57}$	$f_{58}$
$f_{61}$	$f_{62}$	$f_{63}$	$f_{64}$	$f_{65}$	$f_{66}$	$f_{67}$	$f_{68}$
$f_{71}$	$f_{72}$	$f_{73}$	$f_{74}$	$f_{75}$	$f_{76}$	$f_{77}$	$f_{78}$
$f_{81}$	$f_{82}$	$f_{83}$	$f_{84}$	$f_{85}$	$f_{86}$	$f_{87}$	$f_{88}$

Figure 2-10. An 8x8 spatial domain digital image.

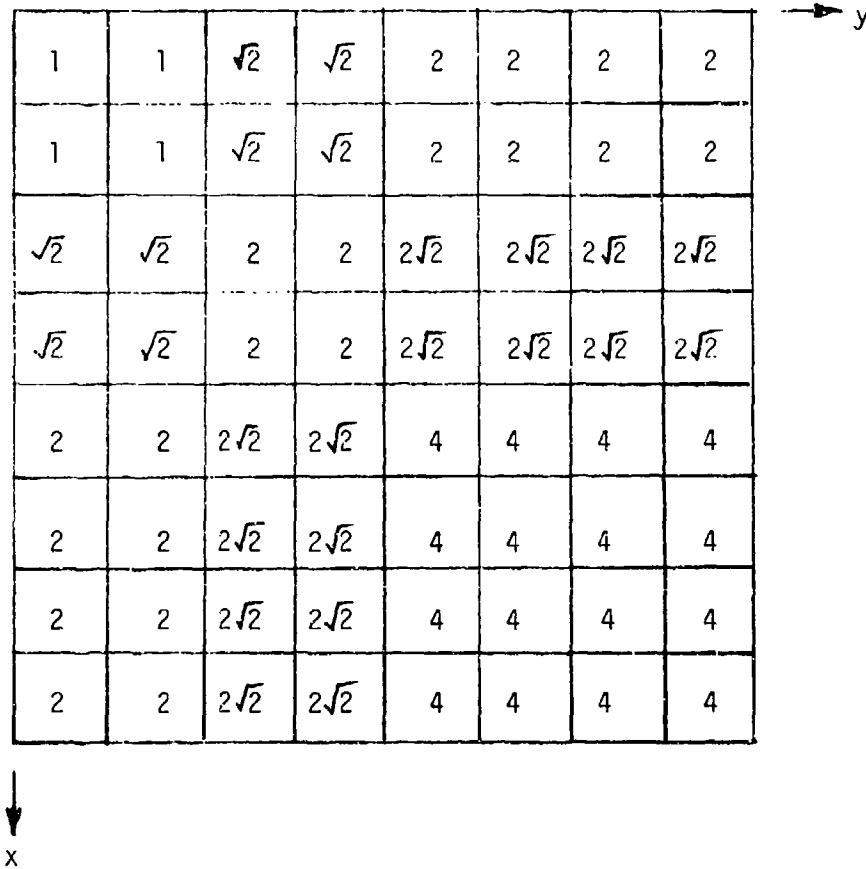


0	0	0	0	1	1	-1	-1
0	0	0	0	1	1	-1	-1
0	0	0	0	-1	-1	1	1
0	0	0	0	-1	-1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

→ y

↓ x

Figure 2-11. The (3, 4) basis plane of Figure 2-9.



An 8x8 grid of weighting factors. The horizontal axis is labeled 'y' with an arrow pointing to the right, and the vertical axis is labeled 'x' with an arrow pointing downwards. The grid contains numerical values that increase from the top-left corner towards the bottom-right corner.

1	1	$\sqrt{2}$	$\sqrt{2}$	2	2	2	2
1	1	$\sqrt{2}$	$\sqrt{2}$	2	2	2	2
$\sqrt{2}$	$\sqrt{2}$	2	2	$2\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$
$\sqrt{2}$	$\sqrt{2}$	2	2	$2\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$	$2\sqrt{2}$
2	2	$2\sqrt{2}$	$2\sqrt{2}$	4	4	4	4
2	2	$2\sqrt{2}$	$2\sqrt{2}$	4	4	4	4
2	2	$2\sqrt{2}$	$2\sqrt{2}$	4	4	4	4
2	2	$2\sqrt{2}$	$2\sqrt{2}$	4	4	4	4

Figure 2-12. The weighting factors for the 8x8 transform-domain image.

Other coefficients are computed in a similar way by repeating the process of correlation with the corresponding basis plane.

The Haar transform can be computed efficiently by using a fast Haar transform (FHT). A flow diagram for computing an 8-point fast Haar transform is shown in Figure 2-13. In the diagram the input sequence numbers are represented by  $t_i$  and the transformed coefficients by  $F_i$ . Each node represents a summing junction and multiplication by constants at various stages of the computation is indicated by  $-1$  or  $\sqrt{2}$  or  $2$  along the branches. A two dimensional Haar transform is then computed by repeated application of the one dimensional FHT.

#### 1. Modified and Rationalized Haar Transforms

Since some of the elements in the Haar matrix are irrational numbers (powers of  $\sqrt{2}$ ), a variation of the transform called the modified Haar transform (MHT), which does not require a multiplier, is more attractive for digital implementation. A modified Haar matrix whose elements are either 1,  $-1$ , or 0 is shown in Figure 2-14.

Another variation of the Haar transform that does not involve numbers which are powers of  $\sqrt{2}$  is the rationalized Haar transform (RHT). A rationalized Haar matrix has elements which are either 1,  $-1$ , 0 or integer powers of 2 as shown in Figure 2-15. The multiplication by powers-of-two is easily implemented as binary shifts.

All the transform techniques described possess a fast computational algorithm when the order of the transform being computed is an integer power-of-two. The computational requirements of the FHT, MHT, and RHT are compared with several other well known transforms in Table 11.

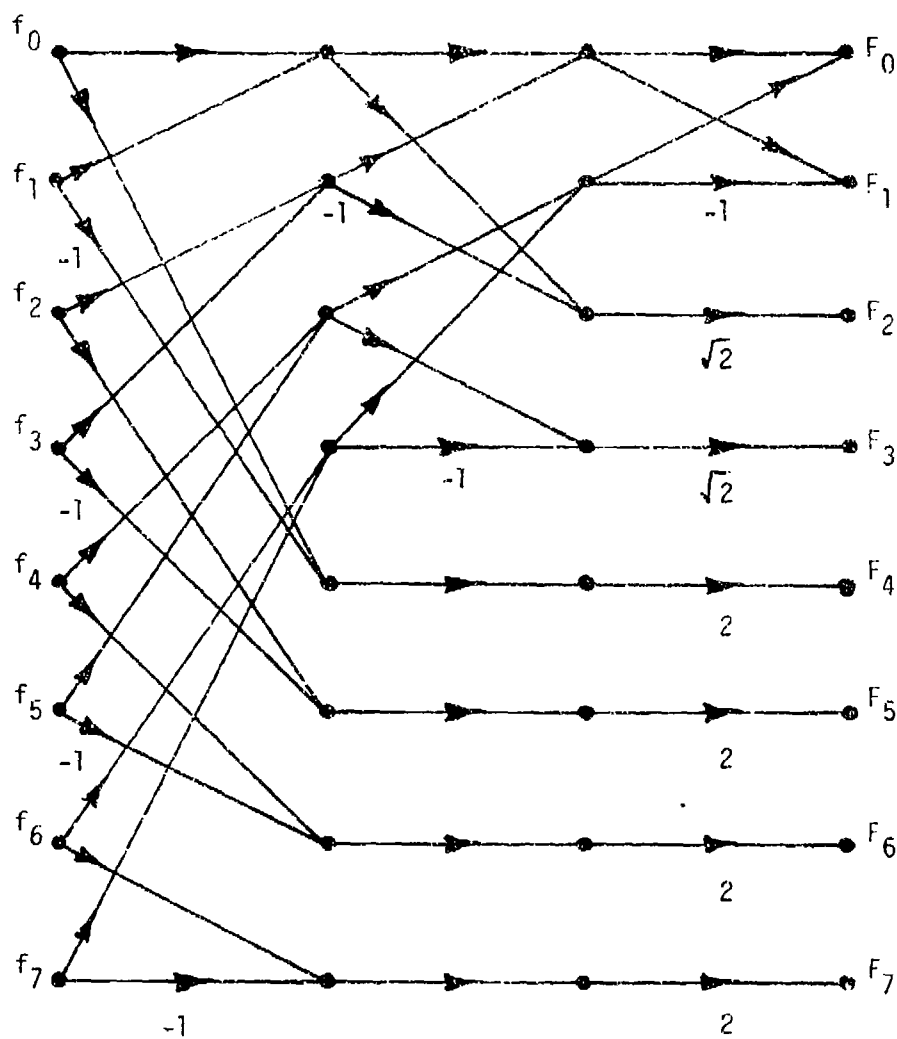


Figure 2-13. A Flow Diagram for an 8 Point Fast Haar Transform (FHT).

$$H_m = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Figure 2-14. An 8 x 8 modified-Haar matrix.

$$H_r = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 2 & -2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & -2 & -2 \\ 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & -4 \end{bmatrix}$$

Figure 2-15. An 8 x 8 rationalized-Haar matrix.

Both modified Haar and rationalized Haar transforms possess the following important properties of the Haar transform:

- Images can be reconstructed by taking the inverse Haar transform
- They possess a fast computational algorithm
- Transform coefficients are locally as well as globally sensitive

Table 11. Computational requirements of various two-dimensional transforms.

Image Size	$N \times N$ (N is an integer power of 2)	8x8	16x16	32x32
Fast Haar Transform (FHT), (MHT), (RHT)	$4N(N-1)$ Additions	$224^+$	$960^+$	$3968^+$
Fast Fourier Transform (FFT)	$2N^2 \log_2 N$ Additions $2N^2 \log_2 N$ Multiplications	$384^+$ $384^x$	$2048^+$ $2048^x$	$10240^+$ $10240^x$
Fast Walsh/Hadamard Transform (WHT)	$2N^2 \log_2 N$ Additions	$384^+$	$2048^+$	$10240^+$
Fast Hadamard Haar Transform (HHT)	$2N(3N-4)$ Additions	$320^+$	$1408^+$	$4888^+$

+ indicates addition

x indicates multiplication

- Transforms are capable of representing a digital image with a few constituent elements in the transform domain to a high degree of accuracy

In a transform domain image, the low sequency large amplitude components represent the image background and the high sequency large amplitude components represent the image edges. Since the high sequency components of the modified Haar transform are weighted less than the corresponding components of the Haar transform, the MHT can be considered as a kind of low-pass filter with a smoothing effect. The RHT, because the high sequency components are weighted more, tends to represent well defined edges in the image domain as edges in the transform domain with large component values. Therefore, RHT produces an enhancing effect on the image edges.

## 2. Selection of Haar Coefficients as Features

One of the several criteria for selection of features for matching purposes is to select those transform coefficients which have the largest variation or variance across image classes. The computation of the variance of the coefficients of the transformed image of size  $K \times K$  using a FLIR image of size  $M \times N$  requires  $(M-K+1)(N-K+1)$  Haar transforms to be computed to include all possible image classes. Since it is extremely time consuming to compute  $(M-K+1)(N-K+1)$  Haar transforms for large  $M$  and  $N$ , only a small number of equally spaced subimages of size  $K \times K$  are selected from the FLIR image and used for the computation of the coefficient variances. The computation of the transform coefficient variances for reduced low resolution FLIR images is explained in the following steps:



1. Four hundred equally spaced subimages of size  $32 \times 32$  are selected from the reduced low resolution FLIR image of size  $406 \times 313$ .
2. The Haar transform of each of the subimages selected in Step 1 is computed.
3. Steps 1 and 2 are repeated for four different low resolution FLIR images, thus yielding a total of 1600 transformed images.
4. Variance of each transform coefficient is then computed over the 1600 transformed images of Step 3.
5. Steps 1 through 4 are repeated using modified Haar transform and rationalized Haar transform.

The variance distribution of the 50 transform coefficients having the largest variation across image classes is shown in Figure 2-16 for the Haar, modified Haar and rationalized Haar transformed images. The locations of the 64 largest variance transform coefficients for the Haar, modified Haar and rationalized Haar transformed images are shown in Figures 2-17, 2-18 and 2-19, respectively. In these figures, "1" indicates the location of the coefficient with the largest variance, "2" indicates the location of the coefficient with the second largest variance and so on. The transform coefficient variances for reduced high resolution FLIR images are computed in a similar way. The 1600 subimages of size  $32 \times 32$  are chosen by selecting 100 equally spaced subimages from each of the 16 different reduced high resolution FLIR images of size  $98 \times 78$ . The distribution of the transform coefficients with the 50 largest variances is shown in Figure 2-20 and the locations of the 64 largest variance transform coefficients for the Haar, modified Haar and rationalized Haar

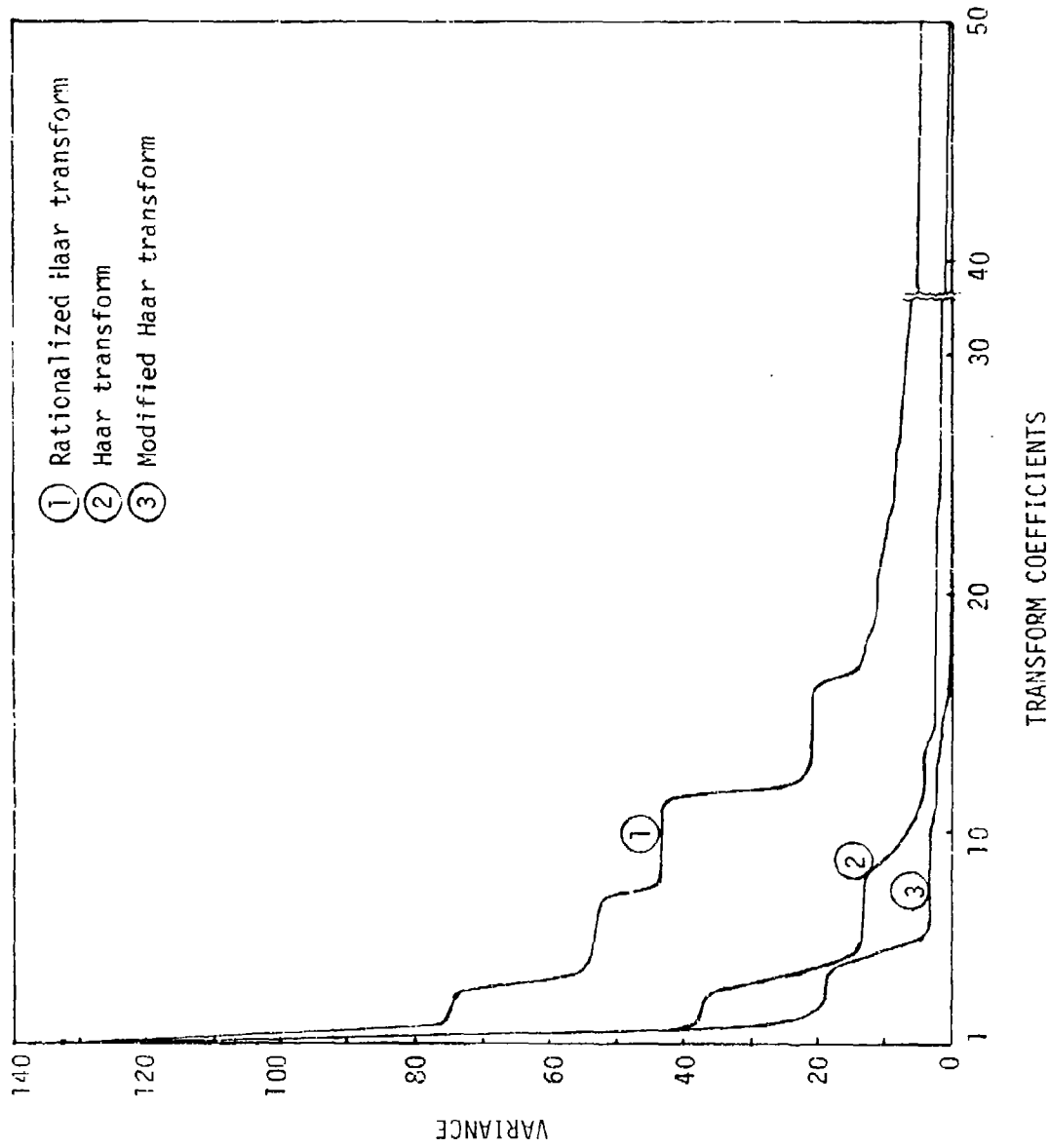


Figure 2-16. Transform coefficient variance distribution for low resolution FLIR images.

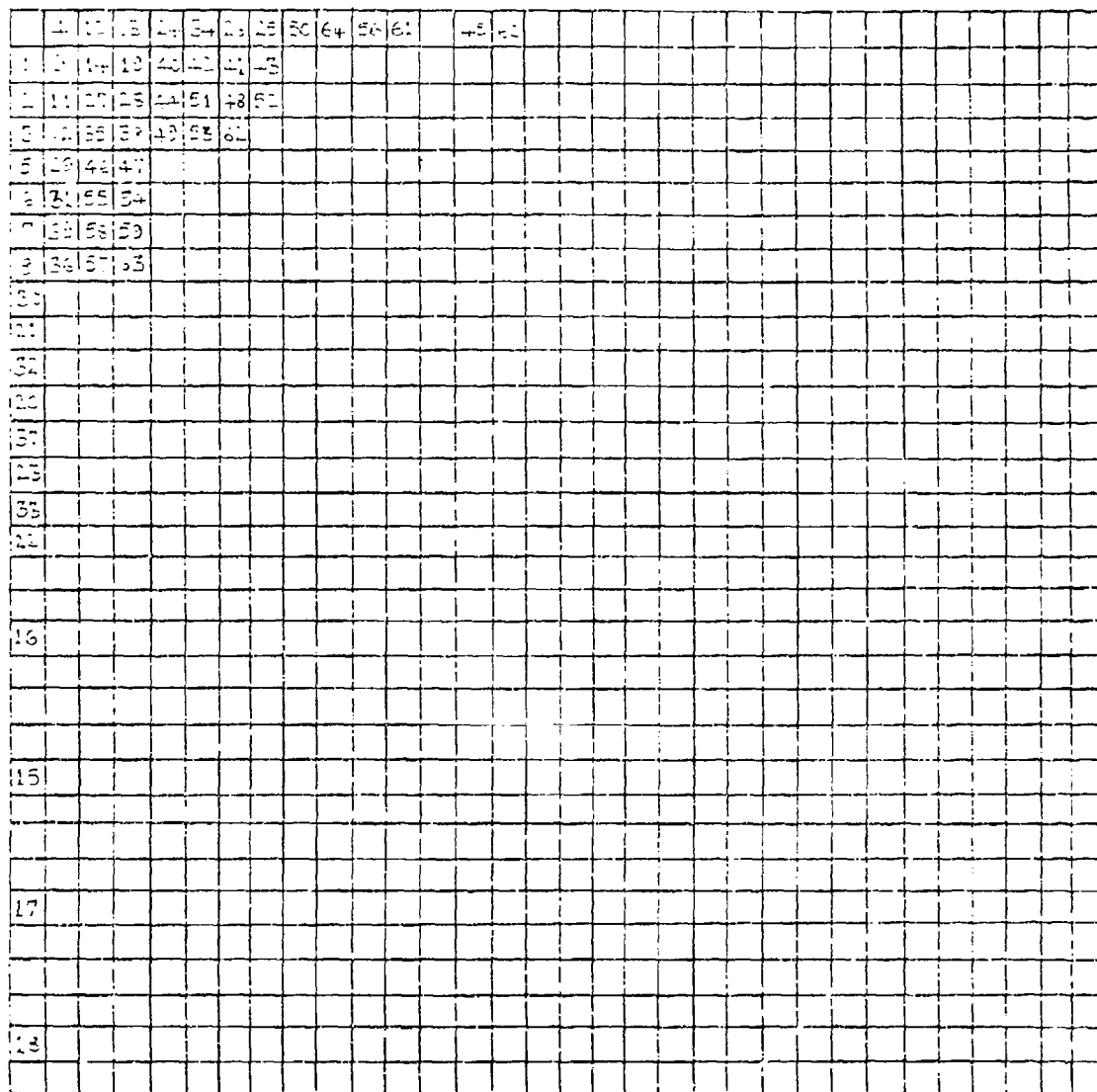


Figure 2-17. Location of the 64 Haar transform coefficients having largest variances (Low resolution FLIR images).

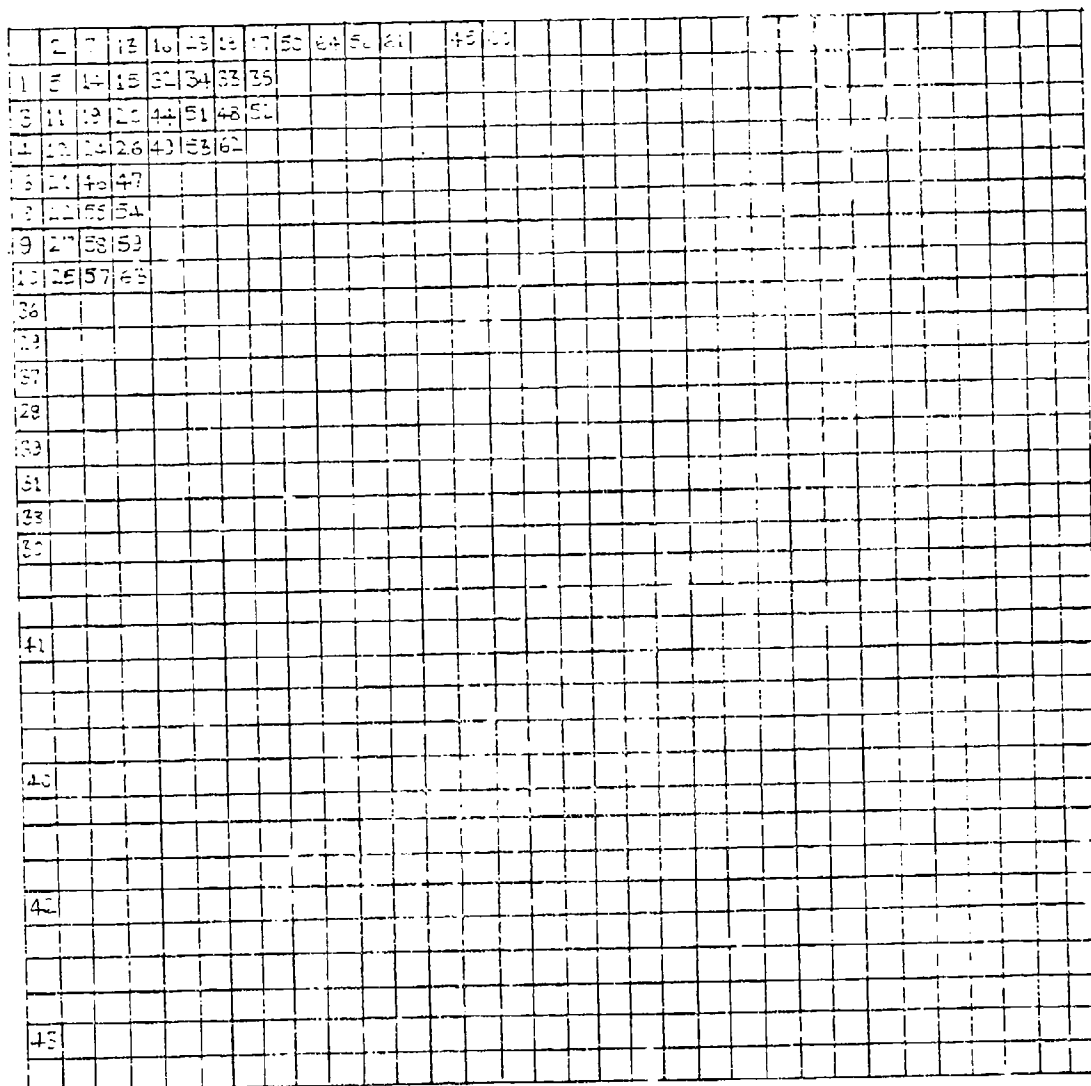


Figure 2-18. Location of the 64 modified Haar transform coefficients having largest variances (Low resolution FLIR images).

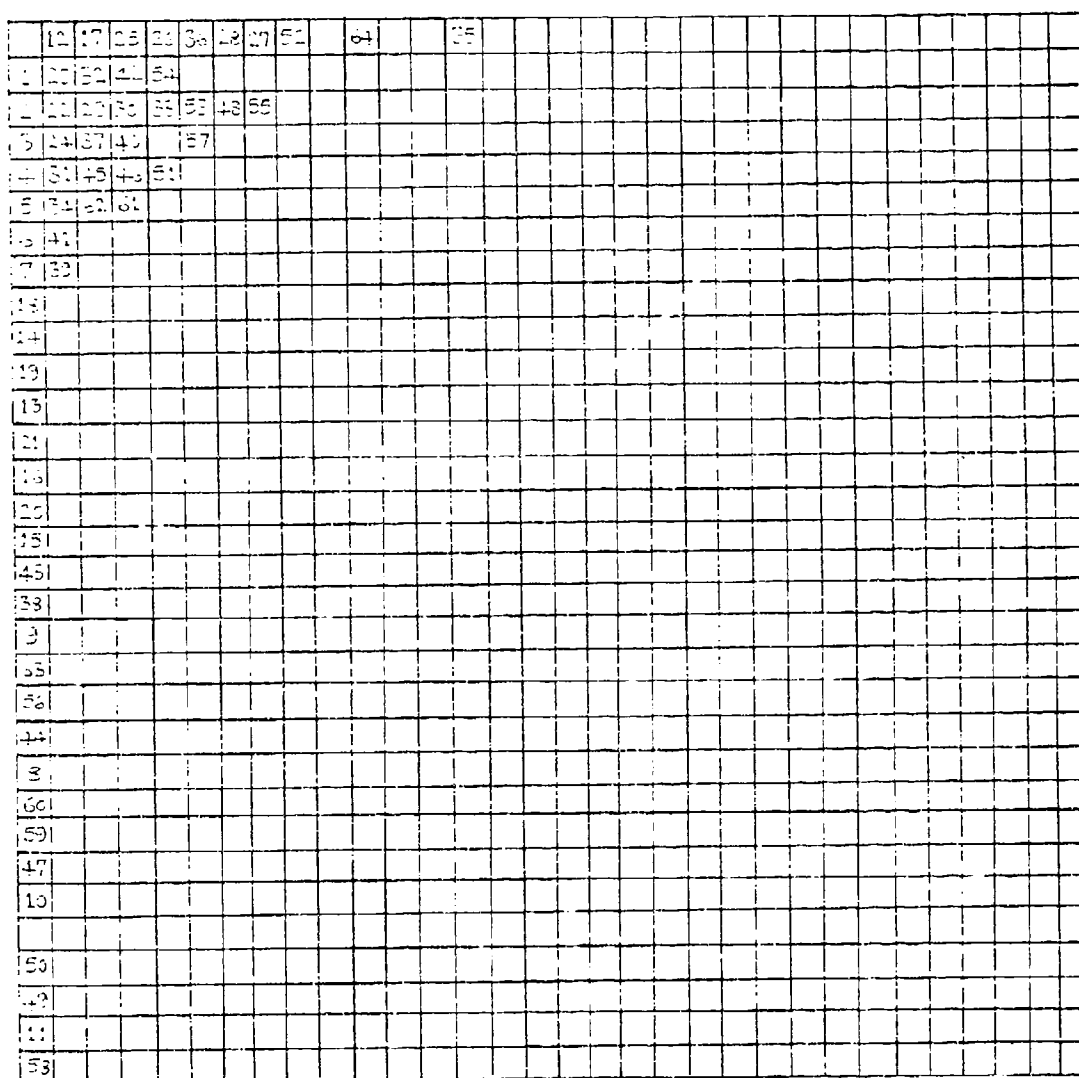


Figure 2-19 . Location of the 64 rationalized Haar transform coefficients having largest variances (Low resolution FLIR images).

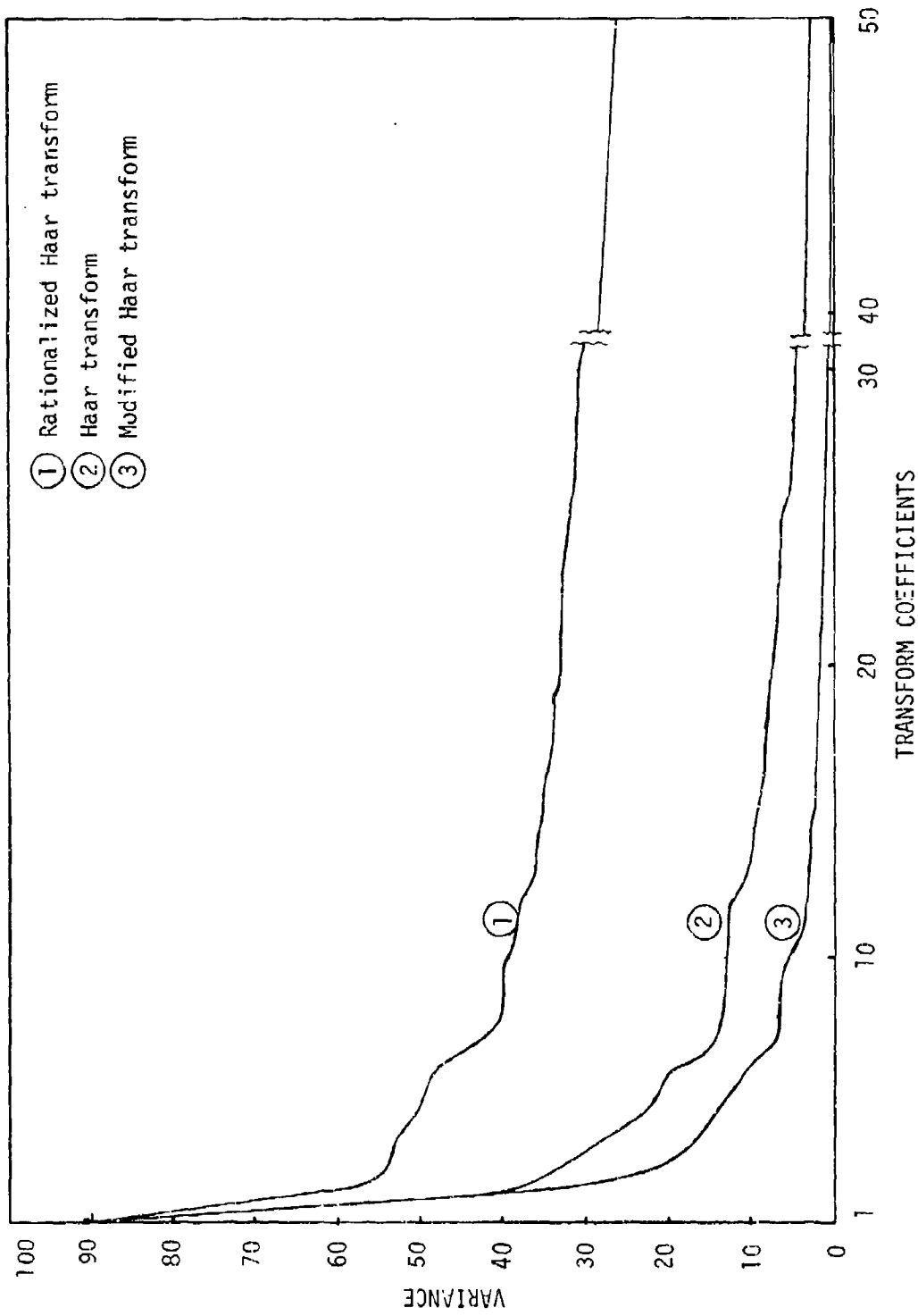


Figure 2-20 . Transform coefficient variance distribution for high resolution FLIR images.

transformed images are shown in Figures 2-21, 2-22, and 2-23, respectively.

From Figures 2-16 through 2-23 the following conclusions are made:

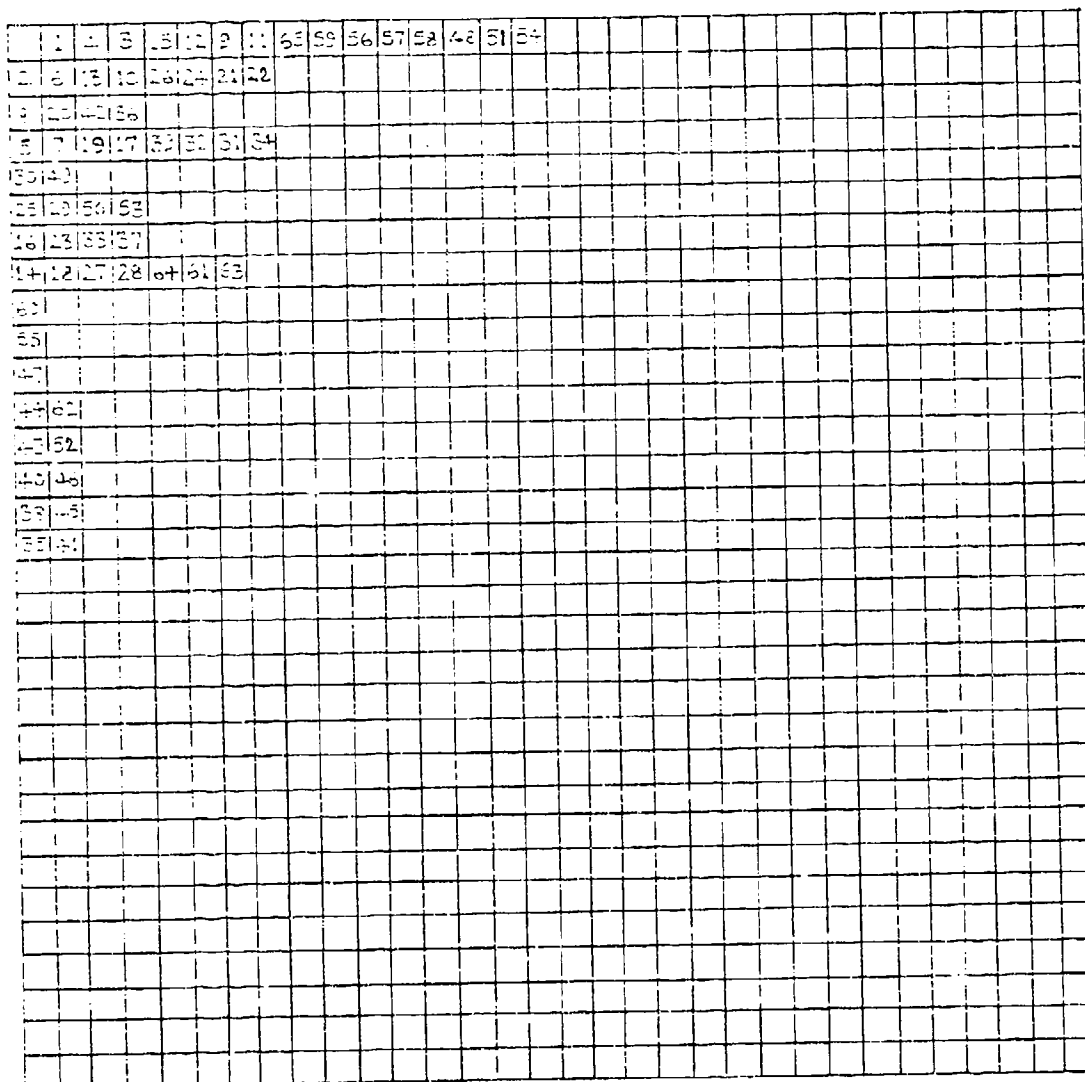
1. A few transform coefficients (about 30) have relatively large variances.
2. The low sequency components of the transformed image have higher variances compared to those of the high sequency components for the scenes used.

The above two facts can be used to reduce the dimensionality of the feature vector while retaining the prominent features of the image for matching purposes.

### 3. Derivation of the Computational Requirements for Multiple Image Registration

Computation of the modified and rationalized Haar transform requires only addition and subtraction and possesses a fast computational algorithm when the size of the image is an integer power of two. The two dimensional fast MHT or RHT of a digital image of size  $N \times N$  requires  $4N(N-1)$  additions. For a search area of size  $M \times N$  and a reference image (window) of size  $K \times K$  the number of additions and multiplications required to implement the algorithm is derived in the following steps:

1. Computation of the fast MHT or RHT of a digital image of size  $K \times K$  requires  $4K(K-1)$  additions. Since there are  $(M-K+1)(N-K+1)$  subimages in a search area of size  $M \times N$ , a total of  $4K(K-1)(M-K+1)(N-K+1)$  additions is required.
2. To compute the Haar transform of a window of size  $K \times K$ , an additional  $4K(K-1)$  additions are needed.





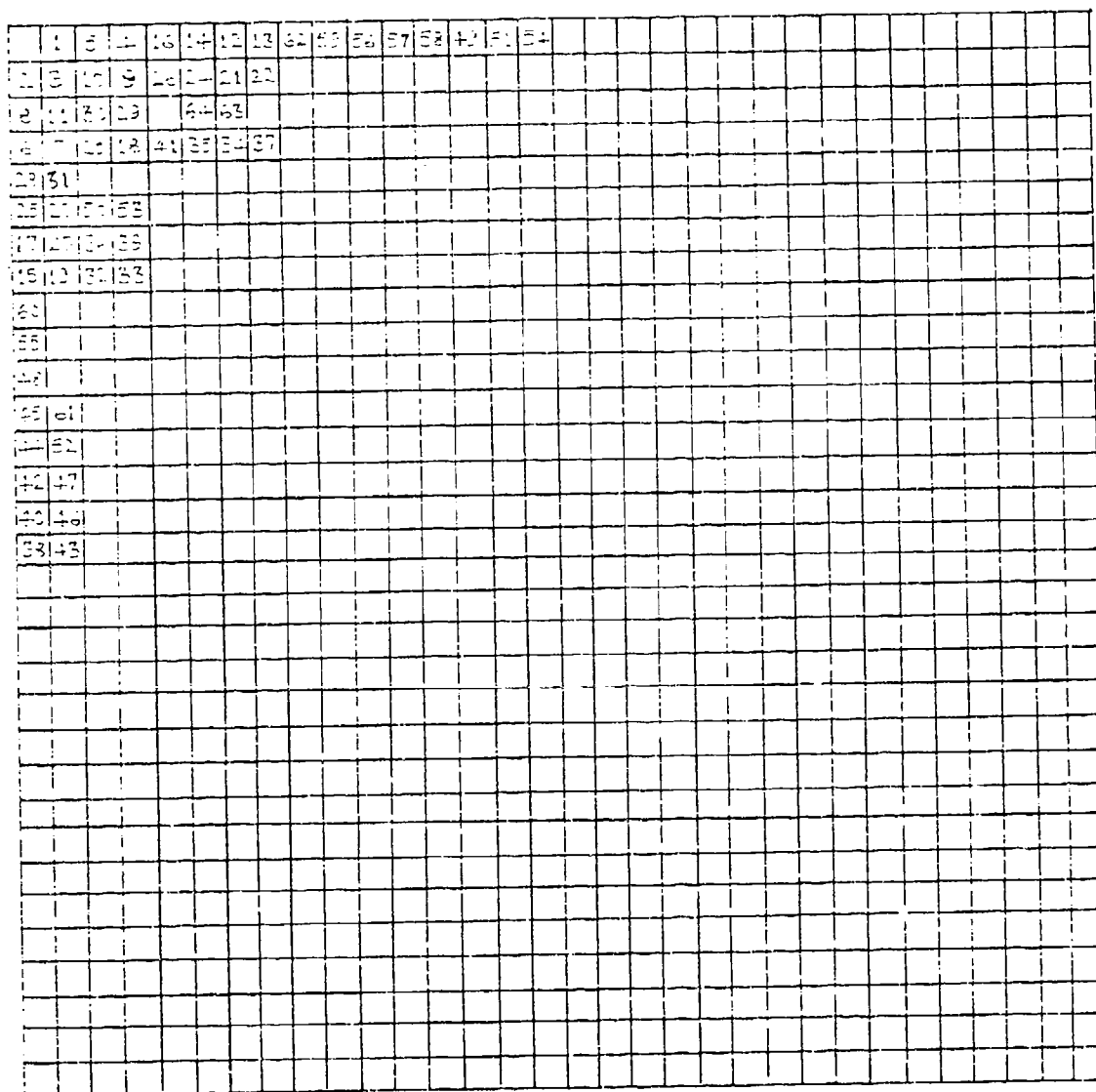


Figure 2-22. Location of the 64 modified Haar transform coefficients having largest variances (High resolution FLIR images).

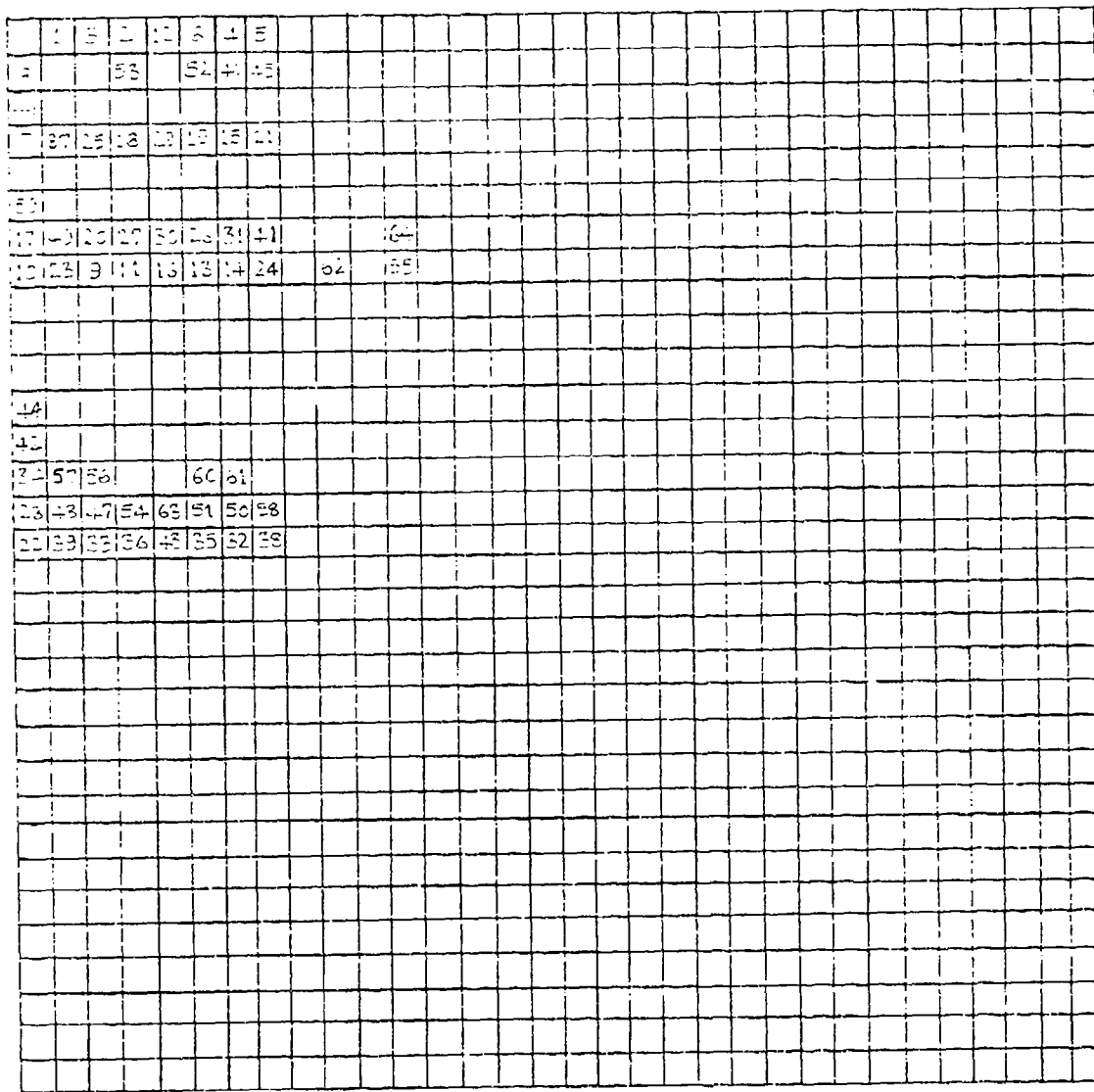


Figure 2-23. Location of the 64 rationalized Haar transform coefficients having largest variances (High resolution FLIR images).

3. The correlation surface is generated by correlating the reference feature vector with each of the  $(M-K+1)(N-K+1)$  feature vectors of the subimage using the relation

$$C_{i,j} = \frac{(V_W^T V_{i,j}) |(V_W^T V_{i,j})|}{||V_W||^2 ||V_{i,j}||^2}, \text{ for all } i,j$$

Computation of  $C_{i,j}$  for feature vectors of dimension  $P$  requires  $(3P-3)$  additions and  $(3P+5)$  multiplications. The generation of the correlation surface, therefore, requires a total of  $(M-K+1)(N-K+1)$   $(3P-3)$  additions and  $(M-K+1)(N-K+1)(3P+5)$  multiplications.

4. Assuming there are  $n$  windows, the multiple image registration process requires Steps 2 and 3 to be performed  $n$  times. Therefore,

$$\begin{aligned} \text{Total number of additions} &= 4K(K-1)(M-K+1)(N-K+1) \\ &\quad + n[4K(K-1) + (3P-3)(M-K+1)(N-K+1)] \end{aligned}$$

$$\text{Total number of multiplications} = n(M-K+1)(N-K+1)(3P+5)$$

$$\text{For } M = 240, N = 256, K = 32, \text{ and } P = 64$$

$$\text{Total number of additions} = 186595200 + 8891693 n$$

$$\text{Total number of multiplications} = 9263925 n$$

The number of multiplications required is negligible when compared to the number of additions. The number of additions and multiplications which are required to implement correlation, SSDA, moments method and Haar transform technique for various values of  $n$ , are shown in Figures 2-24 and 2-25 respectively. Assuming that multiplication, division and squaring operations are equivalent and that the time required to perform one real multiplication is three times that required to perform one real addition

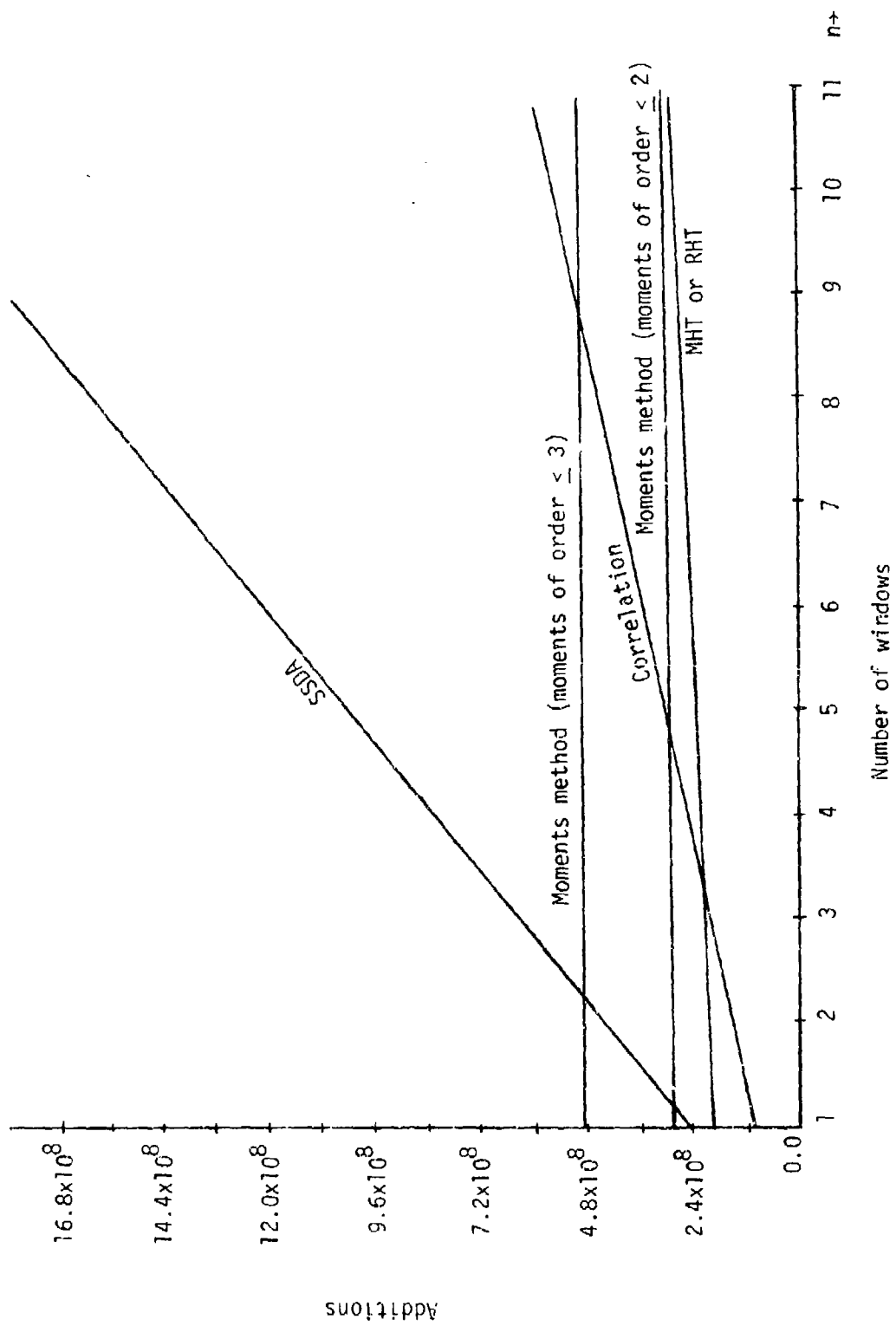


Figure 2-24. Number of additions to register  $n$  windows.

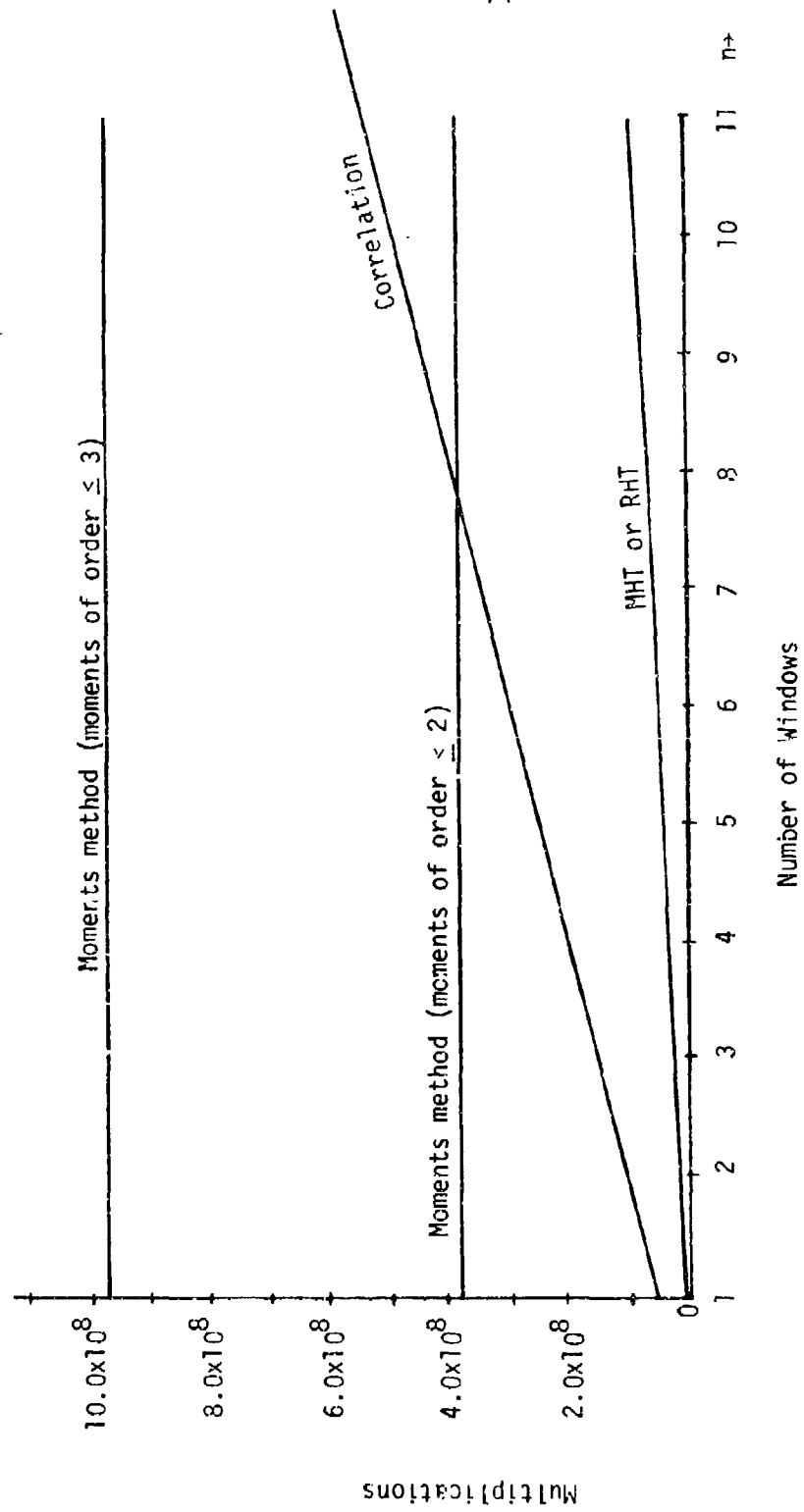


Figure 2-25. Number of multiplications to register  $n$  windows.

(IBM 370), the total number of equivalent real additions required to implement the various methods is shown in Figure 2-26.

#### 4. Simulation of Haar Transforms

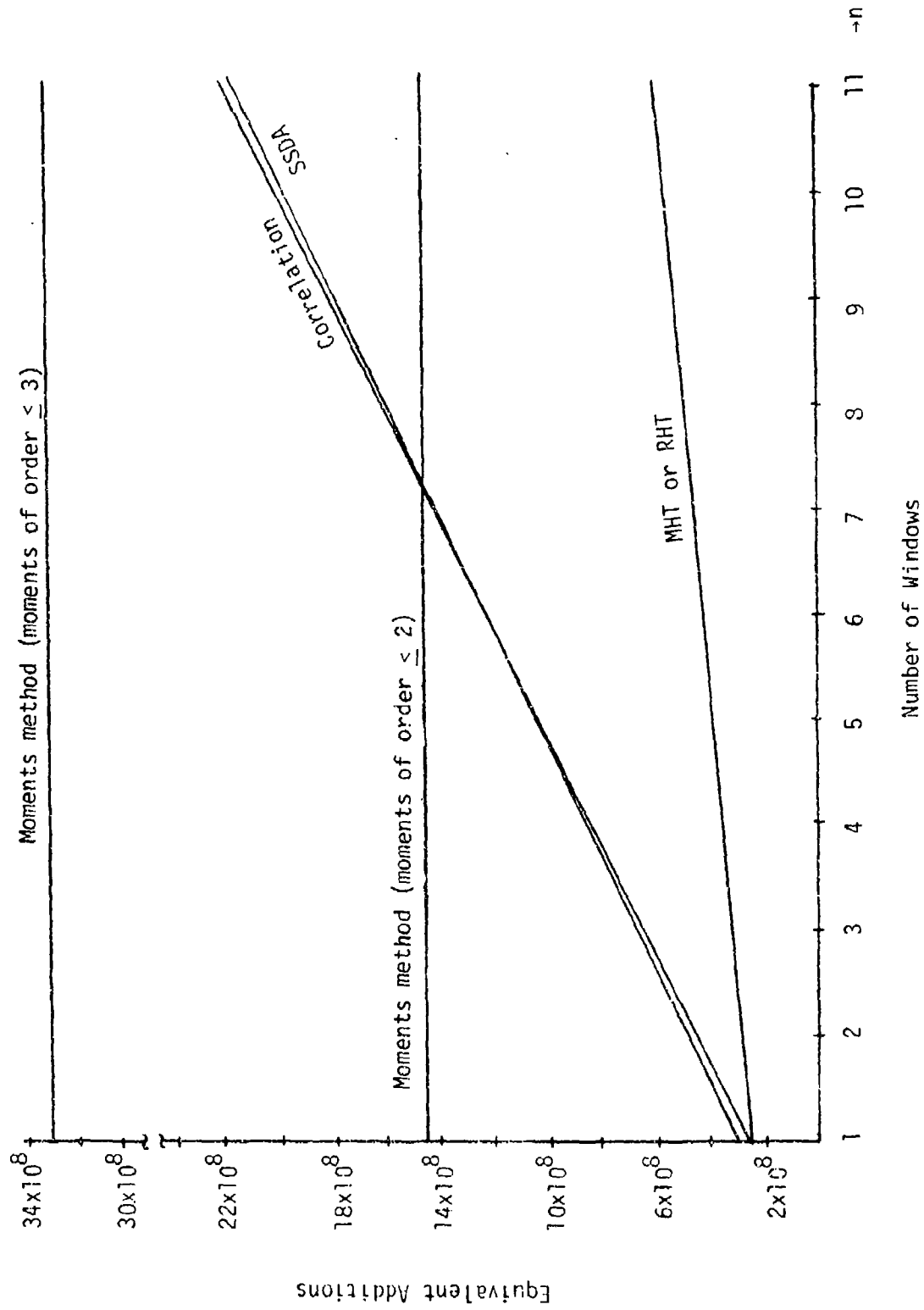
Haar transform and the variations of the Haar transform tend to result in uncorrelated coefficients in the transformed image (to decompose the image into a relatively small number of large amplitude coefficients representing most of the image information) of an originally highly correlated image. These uncorrelated coefficients are expected to provide a set of features which are more efficient for matching purposes than a similar number of features in the originally highly correlated digital image. However, the performance and accuracy of the image registration algorithm depends largely on the selection of the transform-coefficients for matching purposes. Several simulations were run to study the efficiency and performance of the image matching technique using Haar transform-coefficient feature vectors. Reduced low and high resolution FLIR images described earlier are used in the simulation. The registration technique and the simulation results are presented below for all the cases considered.

Case 1: The registration technique is described in the following steps.

Step 1: A reference image  $W$  of size  $32 \times 32$  is selected from the reduced high resolution FLIR image so as to include prominent features of the image.

Step 2: The two dimensional Haar transform of  $W$  is computed by using the fast Haar transform (FHT).

Step 3: A 64-dimensional feature vector  $V_W$  consisting of the 64 highest coefficients of the transformed image is formed after setting the

Figure 2-26. Number of equivalent additions to register  $n$  windows.

(1,1) element of the transformed image to zero. Since the (1,1) element represents the mean of the pixel values of W, setting this element to zero simply ignores the image mean values which may be different. All other coefficients are insensitive to the differing means since they are the sums and difference of pixel values (i.e., subtracting the mean value before computing these coefficients will not change the coefficient values).

Step 4: In order to reduce simulation computation time, a subarray of size 100 x 180 which includes the target area of interest is extracted from the 406 x 313 reduced low resolution FLIR image (search area).

Step 5: The Haar transform of each subimage of size 32 x 32 from the search area is computed and a 64-dimensional feature vector,  $V_{i,j}$ , is formed for each subimage in such a way that the elements of  $V_{i,j}$  and  $V_W$  have common locations in the transformed images.

Step 6: The cosine of the angle (phase correlation) between the two feature vectors  $V_W$  and  $V_{i,j}$  is computed using the relation

$$C_{i,j} = \frac{V_W^T V_{i,j}}{\|V_W\| \|V_{i,j}\|}, \text{ for all } i,j$$

Step 7:  $C_{i,j}$  is expected to have a maximum at the registration point.

Case 2: In Case 1, the reference images are chosen to include prominent features of the image and was not necessarily taken from the center of the field of view of the reduced high resolution FLIR image. Case 2 is identical to Case 1, except that the reference



images are chosen from the center of the field of view of the reduced high resolution FLIR image.

Case 3: In this case, reference arrays of size  $32 \times 64$  are chosen from the center of the field of view of the reduced high resolution FLIR image. Since the computation of the FHT requires the image array to be square, the  $32 \times 64$  reference array is processed as two separate blocks of size  $32 \times 32$  each. The registration algorithm is described below.

Step 1: The selected reference image of size  $32 \times 64$  is divided into two subimages  $W_1$  and  $W_2$  of size  $32 \times 32$  each.

Step 2: Two dimensional Haar transforms of  $W_1$  and  $W_2$  are computed separately using the FHT.

Step 3: A 128-dimensional feature vector  $V_W$  consisting of the 64 highest coefficients of the transformed image of  $W_1$  and 64 highest coefficients of the transformed image of  $W_2$  is formed. The (1,1) elements of the transformed images, which represent the means of the pixel values of  $W_1$  and  $W_2$ , are not considered in the selection of the highest coefficients.

The remaining procedure is as in Steps 4 through 7 of Case 1, except that the subimages are of size  $32 \times 64$  and the feature vector  $V_{i,j}$  of subimage  $S_{i,j}$  is of dimension 128 as in Step 3.

Case 4: In all the cases considered above, digital image registration is accomplished by matching feature vectors whose elements form a set of predetermined number of highest transform coefficients. Another criterion for the selection of features for matching purposes is to select those transform coefficients which have the

largest variation or variance across image classes. Observation of Figures 2-16 through 2-23 indicates that most of the transform coefficients with largest variance are the low frequency components. Therefore, this case is an effort to accomplish image registration using features consisting of the low frequency components.

Reference arrays of size  $32 \times 32$  are chosen from the center of the field of view of the reduced high resolution FLIR image. The feature vector of the reference image,  $V_W$ , is formed by selecting the 63 coefficients from the  $8 \times 8$  array at the top left corner of the transformed image excluding the (1,1) element, which represents the mean of the pixel values of  $W$ . This  $8 \times 8$  array of transform coefficients can easily be computed without actually computing the remaining coefficients by first reducing the resolution of the reference array of size  $32 \times 32$  by a factor of 4 to form a reduced reference array of size  $8 \times 8$  and then computing the Haar transform of the reduced array. The 63-dimensional feature vector,  $V_{i,j}$ , for each subimage  $S_{i,j}$  of size  $32 \times 32$  from the search area of size  $120 \times 240$  (extracted from the  $406 \times 313$  reduced low resolution FLIR image) is also computed in a similar manner. The remaining procedure of accomplishing image registration is as in Case 1.

Case 5: This case is similar to Case 4 except that rationalized Haar transform (RHT) is used to compute the transform coefficients.

Case 6: Here only a subset of 32 coefficients from the 63 coefficients selected in Case 5, is used to form the feature vectors. The rest of the procedure is as in Case 5. The feature vectors of the reference and subimages of the search area are formed as follows.

1. An  $8 \times 8$  array of transform-coefficients at the upper left corner of the transformed  $32 \times 32$  digital image is computed as explained in Case 4.
2. Sixteen coefficients of the  $4 \times 4$  array at the bottom left corner and sixteen coefficients of the  $4 \times 4$  array at the upper right corner of the  $8 \times 8$  array of transform coefficients computed in Step 1 are used to form the 32-dimensional feature vector.

Case 7: Reference arrays of size  $32 \times 64$  are chosen from the center of the field of view of the reduced high resolution FLIR image and are processed as two separate blocks of size  $32 \times 32$  each. The registration algorithm is described below.

- Step 1: The selected reference image of size  $32 \times 64$  is divided into two subimages  $W_1$  and  $W_2$  of size  $32 \times 32$  each.
- Step 2: The resolution of  $W_1$  and  $W_2$  is reduced by a factor of 4 and the Haar transform of each reduced subimage is computed to form the reduced transformed subimages  $W_{1R}$  and  $W_{2R}$  of size  $8 \times 8$  each.
- Step 3: A 126-dimensional feature vector,  $V_W$ , consisting of the coefficients of  $W_{1R}$  and  $W_{2R}$ , excluding the  $(1,1)$  elements, is formed.

- Step 4: For each subimage  $S_{i,j}$  from the search area of size  $120 \times 240$ , a 126-dimensional feature vector  $V_{i,j}$  is computed according to Steps 1 through 3.
- Step 5: The feature vectors  $V_W$  and  $V_{i,j}$  are correlated to determine the true registration point.
- Case 8: Similar to Case 7 except that in Step 2, the modified Haar transform (MHT) is used to compute the transform coefficients.
- Case 9: Similar to Case 7 except that in Step 2 the RHT is used to compute the transform coefficients.
- Case 10: As in Case 7, with the following changes in Steps 3 and 4.
- Step 3: Sixteen coefficients of the  $4 \times 4$  array at the bottom left corner and sixteen coefficients of the  $4 \times 4$  array at the top right corner are selected from both  $W_{1R}$  and  $W_{2R}$  to form a 64-dimensional feature vector  $V_W$ .
- Step 4: For each subimage  $S_{i,j}$  from the search area of size  $120 \times 240$ , a 64-dimensional feature vector  $V_{i,j}$  is computed according to Steps 1 through 3.

Simulation results of the feature matching registration technique using Haar, modified and rationalized Haar coefficients for all the cases considered above are presented in Table 12 and summarized in Table 13. From observation of the results summarized in Table 13, the following conclusions can be made. The modified Haar transform has the poorest performance and is not suitable for extracting features for matching purposes. One possible reason for its poor performance is the suppression of the high frequency components of the transformed image, which represent most of the edge information. The RHT, which has an enhancing effect on

Table 12. Simulation results of the feature matching image registration technique using Haar-coefficients.

Scenes		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
Position A	Target #1	1	1	-	1	1	1	-
	Target #2	1	X	1	X	X	1	1
	Target #3	2	X	X	X	2	1	X
	Target #4	X	X	X	X	X	2	X
Position B	Target #1	X	X	X	X	X	X	X
	Target #2	1	1	1	X	1	X	2
	Target #3	3	1	2	2	1	X	2
	Target #4	3	X	X	3	2	3	X
Position C	Target #1	1	1	1	2	X	X	1
	Target #2	1	X	1	X	1	1	1
	Target #3	1	1	1	1	1	1	1
	Target #4	1	1	1	1	1	1	1
Position D	Target #1	1	X	X	X	X	X	X
	Target #2	1	X	2	X	3	2	2
	Target #3	2	X	X	X	X	X	X
	Target #4	X	2	4	1	1	1	X

Table 12. Continued

Scenes		Case 8	Case 9	Case 10
Position A	Target #1	-	-	-
	Target #2	2	1	1
	Target #3	X	1	1
	Target #4	X	X	X
Position B	Target #1	X	3	2
	Target #2	4	1	1
	Target #3	X	1	X
	Target #4	X	3	4
Position C	Target #1	2	1	1
	Target #2	2	1	1
	Target #3	1	1	1
	Target #4	2	1	1
Position D	Target #1	X	X	X
	Target #2	X	1	1
	Target #3	X	1	2
	Target #4	X	2	3

Table 13. Summary of results in Table 12.

Registration Technique	Registration Point is First Maximum	Registration Point is 2nd, 3rd, or 4th Maximum	Registration Point not is First Four Maxima
Case 1 - FHT, 32x32 re- ference image 64- dimensional feature vector (Reference not from Center of FOV)	9	4	3
Case 2 - FHT, 32x32 re- ference image 64- dimensional feature vector	6	1	9
Case 3 - FHT, 32x64 re- ference image, 128- dimensional feature vector	6	3	6
Case 4 - FHT, 32x32 re- ference image, 63- dimensional feature vector	4	3	9
Case 5 - RHT, 32x32 re- ference image, 63- dimensional feature vector	7	3	6
Case 6 - RHT, 32x32 re- ference image, 32- dimensional feature vector	7	3	6

Table 13. Continued

Registration Technique	Registration Point is First Maximum	Registration Point is 2nd, 3rd, or 4th Maximum	Registration Point not is First Four Maxima
Case 7 - FHT, 32x64 re- ference image, 126- dimensional feature vector	5	3	7
Case 8 - MHT, 32x64 re- ference image, 126- dimensional feature vector	1	5	9
Case 9 - RHT, 32x64 re- ference image, 126- dimensional feature vector	10	3	2
Case 10 - RHT, 32x64 reference image, 64- dimensional feature vector	8	4	3



the edges, performed better than the Haar and modified Haar transforms. The superior performance of RHT makes it more suitable for selection of features for matching purposes for the type of scenes considered. In addition, since the computation of RHT does not involve multiplication by irrational numbers, it can easily be implemented with only adders and subtractors.

#### I. Digital Image Registration Using Walsh/Hadamard Transform

Walsh and Hadamard transforms, like the Haar transform, are orthogonal transforms which convert a digital image in the "spatial-domain" into a "transform-domain" image containing the same information. N-dimensional Walsh and Hadamard transforms, where N is an integer power of two, possess a fast computational algorithm and differ only in the order in which the rows and columns appear in the transform matrices. Since the order in which the rows and columns appear is not important for image matching applications, the term Walsh/Hadamard transform (WHT) is used to denote either transform. The rows of the Walsh/Hadamard matrix are generally ordered with increasing number of sign changes or sequency. The sequency ordered  $8 \times 8$  Walsh/Hadamard matrix is shown in Figure 2-27, and the two dimensional basis plane for an  $8 \times 8$  WHT is shown in Figure 2-28. Walsh/Hadamard transform can be implemented with only additions and subtractions, making it attractive for efficient hardware implementation.

An effort is made to study the performance and accuracy of the feature matching technique based on the WHT coefficients. A few simulations were run for a reference array of size  $32 \times 32$  using a 64-dimensional feature vector. In order to reduce simulation computation time, a subarray

$$W = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Figure 2-27. Sequence ordered 8 x 8 Walsh/Hadamard matrix.

of size 100 x 180 which includes the target area of interest in extracted from the 416 x 313 reduced low resolution FLIR image (search area). The registration algorithm is described below.

#### Algorithm

- Step 1: A reference image  $W$  of size 32 x 32 is selected from the center of the field of view of the reduced high resolution FLIR image.
- Step 2: The resolution of  $W$  is reduced by a factor of 2 to yield a reduced reference image  $W_R$  of size 16 x 16, and the two dimensional WHT is computed by using a fast Walsh/Hadamard transform.
- Step 3: A 64-dimensional feature vector,  $V_W$ , consisting of the 64 highest coefficients of the transformed image  $W_R$  is formed after setting the (1,1) element of  $W_R$  to zero. The (1,1) element represents the mean of the pixel values of  $W$ .
- Step 4: The WHT of each subimage  $S_{i,j}$  of size 32 x 32 from the search area is computed according to Step 2 and a 64-dimensional feature

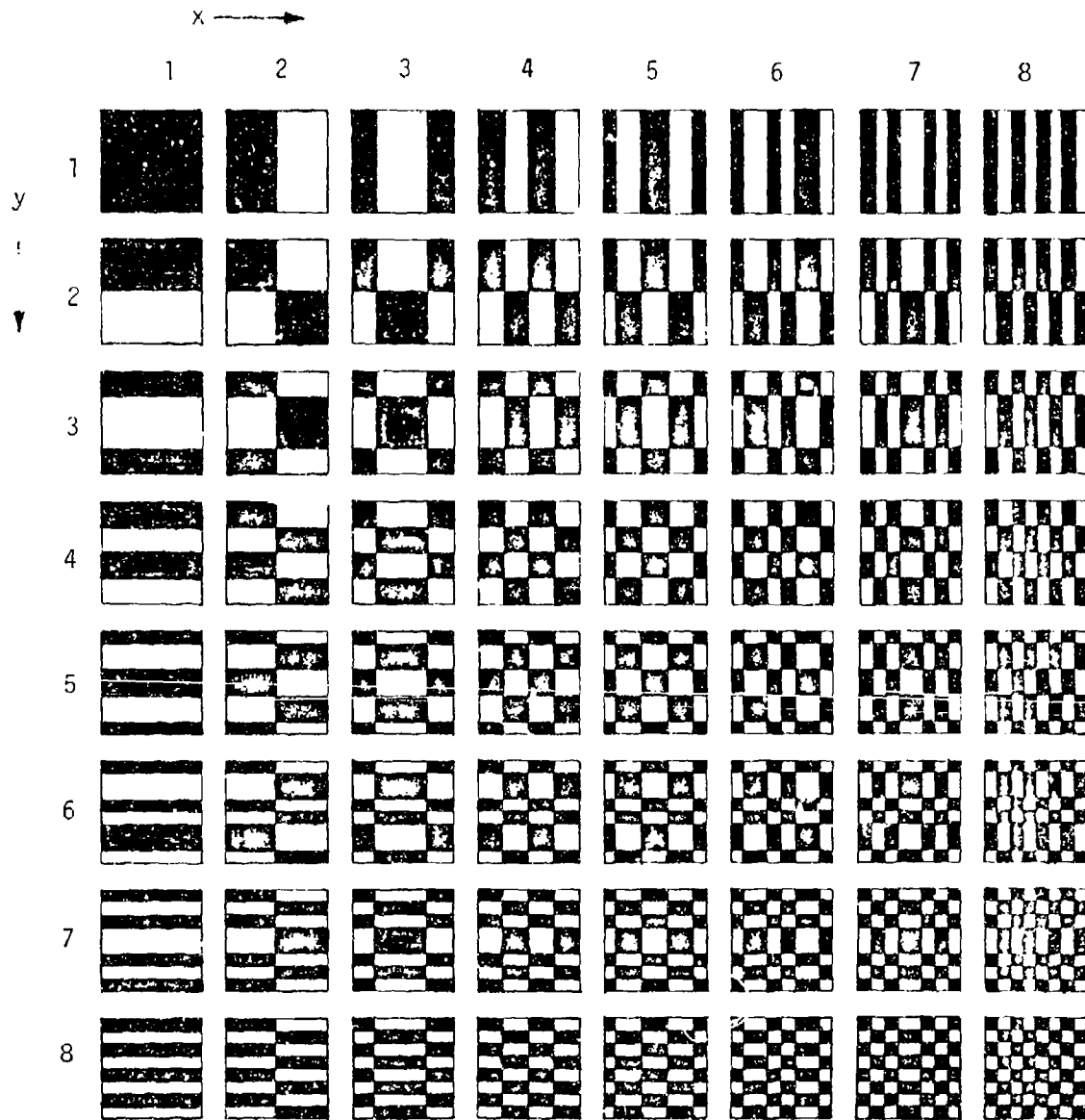


Figure 2-28. Two dimensional basis planes for an 8x8 Walsh/Hadamard transform. Black = +1, white = -1.

vector  $V_{i,j}$  is formed for each subimage in such a way that the elements of  $V_{i,j}$  and  $V_W$  have common locations in the transformed images.

Step 5: The feature vectors  $V_W$  and  $V_{i,j}$  are correlated to find the true registration point.

For the purpose of comparison, simulations were also run by computing Haar transform instead of WHT and the detailed results of the simulation are given in the monthly report of August 1981. A summary of these results is presented in Tables 14 and 15. The results indicate that the Haar transform performs better than the WHT.

Another criterion for selection of features for matching purposes is to select those transform coefficients which have the largest variation or variance across image classes. In order to determine the coefficients with large variances, the variance of each transform coefficient of the Walsh/Hadamard transformed images of size  $32 \times 32$  is computed for both low and high resolution FLIR images over the 1600 transformed images which are selected as outlined in a section on the image registration technique using Haar transform. The variance distribution and the location of the largest variance transform coefficients are essentially the same as that obtained using the Haar transform.

Several simulations were also run with feature vectors consisting of the low sequency components (coefficients with large variance) of the Walsh/Hadamard transformed images. Even though WHT performed comparably with the Haar transform, the performance and registration accuracy of the rationalized Haar transform were superior. It should be noted that each Walsh transform coefficient is a function of all the pixels in the

Table 14. Simulation results of the feature matching technique using WHT and Haar transform coefficients.

Scenes		WHT	Haar
Position A	Target #1	1	1
	Target #2	X	X
	Target #3	X	X
	Target #4	X	X
Position B	Target #1	X	X
	Target #2	1	1
	Target #3	2	1
	Target #4	X	4
Position C	Target #1	1	1
	Target #2	X	X
	Target #3	1	1
	Target #4	1	1
Position D	Target #1	X	X
	Target #2	X	X
	Target #3	X	X
	Target #4	X	3

Table 15. Summary of results in Table 14.

Method	Registration Point is the First Peak	Registration Point is 2nd, 3rd or 4th peak	Registration Point is not in first four peaks
WHT	5	1	10
Haar	6	2	8

original image (i.e., global), whereas only the first four coefficients in the Haar case are global. In other words, the Haar transform provides a domain which is sensitive to both global and local variations of the pixels in an image.

The number of additions and multiplications required to implement the feature matching registration algorithms based on correlation of adjacent pixels, difference between adjacent pixels and the Haar transform for a search area of size  $240 \times 256$  and a reference image (window) of size  $32 \times 32$  are derived in the previous sections. The total number of equivalent real additions required to implement correlation, SSDA, MHT or RHT, correlation of adjacent pixels and the new feature matching method are compared in Figure 2-29. In all the cases, the process of deciding the true registration point is based on a similarity measure which is expressed in terms of the correlation of the 64-dimensional feature vectors of the window and the subimages of the search area.

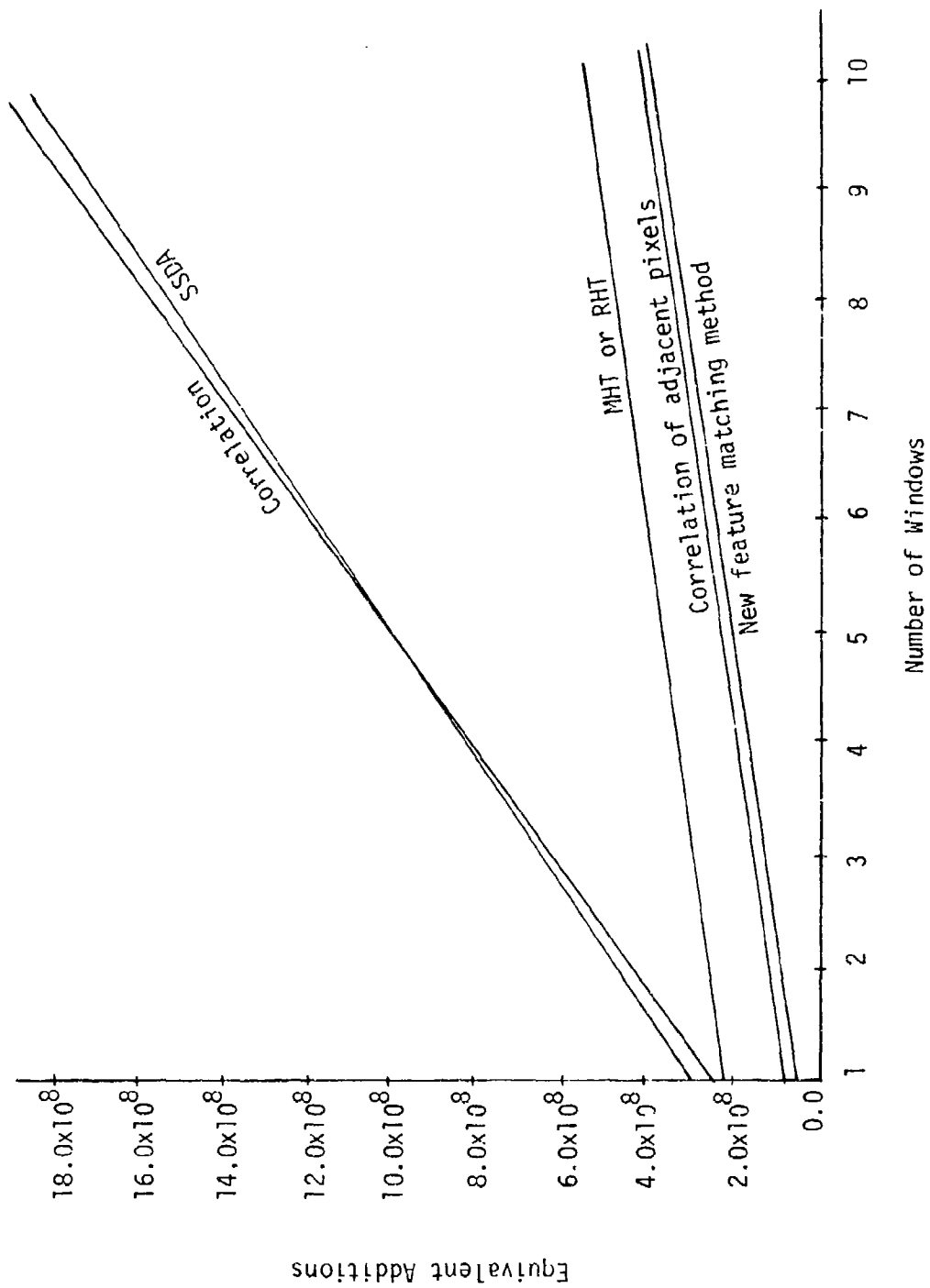


Figure 2-29. Comparison of the arithmetic requirements of various image registration algorithms.



### III. HARDWARE IMPLEMENTATION

Simulation results of various registration algorithms presented in Chapter II indicate that the performance of the moments method and the method based on intraset and interset distances is unsatisfactory for the type of images used for simulation. The other methods seem to have comparable performance to the binary correlation algorithm. However, to accomplish multiple image registration in real time or almost so, image matching algorithms must be implemented using fast hardware. An algorithm which is computationally efficient for software implementation may be complex or even not feasible for hardware implementation. Simple schematics shown in Figures 3-1 through 3-8 are used for comparison of the complexity of hardware implementation of the various methods. Since the moments method and the method based on intraset and interset distances are not accurate for the kind of imagery considered in this work, their hardware implementations are not discussed.

#### A. Standard Cross Correlation Algorithm

No simulations are run using the standard correlation algorithm. However, the standard correlation algorithm is the most popular and proven method of accomplishing digital image registration, and therefore its hardware implementation is discussed in this section. The normalized cross correlation,  $R_k(i,j)$ , between the window  $W_k$  and subimage  $S_{i,j}$  is given by,

$$R_k^2(i,j) = \frac{\left[ \sum_{\ell=1}^K \sum_{m=1}^L W_k(\ell,m) S_{i,j}(\ell,m) \right]^2}{\left[ \sum_{\ell=1}^K \sum_{m=1}^L W_k^2(\ell,m) \right] \left[ \sum_{\ell=1}^K \sum_{m=1}^L S_{i,j}^2(\ell,m) \right]}, \quad (3-1)$$

for  $1 \leq i \leq M-K+1$  and  $1 \leq j \leq N-L+1$ .

$$\text{Let } A_{i,j}^k = \left[ \sum_{\ell=1}^K \sum_{m=1}^L W_k(\ell,m) S_{i,j}(\ell,m) \right]^2 \quad (3-2)$$

$$B_k = \sum_{\ell=1}^K \sum_{m=1}^L W_k^2(\ell,m) \quad (3-3)$$

$$C_{i,j} = \sum_{\ell=1}^K \sum_{m=1}^L S_{i,j}^2(\ell,m) \quad (3-4)$$

Then

$$R_k^2(i,j) = \frac{A_{i,j}^k}{B_k C_{i,j}}, \quad (3-5)$$

for  $1 \leq i \leq M-K+1$  and  $1 \leq j \leq N-L+1$ .

The schematic shown in Figure 3-1 computes  $R_k(i,j)$  in four stages.

Stage 1: In the first stage,  $S_{i,j}^2(\ell,m)$ ,  $W_k^2(\ell,m)$  and the product  $S_{i,j}(\ell,m)W_k(\ell,m)$  are computed for  $\ell = 1, 2, 3, \dots, K$  and  $m = 1, 2, \dots, L$ . This requires a total of  $3KL$  two-input multipliers (three multipliers for each pixel pair).

Stage 2: In the second stage,  $\sqrt{A_{i,j}^k} = \sum_{\ell=1}^K \sum_{m=1}^L S_{i,j}(\ell,m)W_k(\ell,m)$ ,

$B_k = \sum_{\ell=1}^K \sum_{m=1}^L W_k^2(\ell,m)$  and  $C_{i,j} = \sum_{\ell=1}^K \sum_{m=1}^L S_{i,j}^2(\ell,m)$  are computed

using three  $KL$ -input adders.

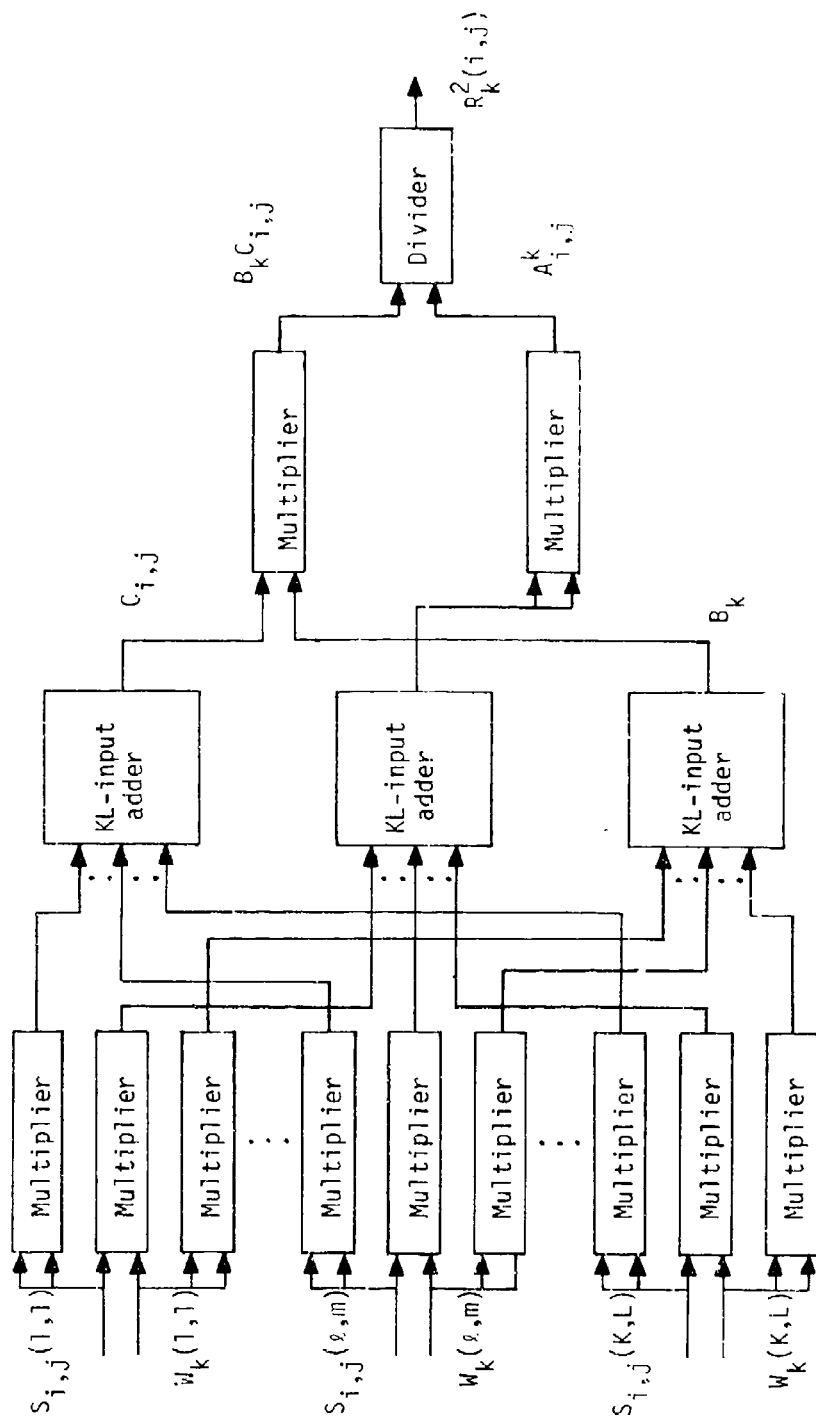


Figure 3-i. Schematic for correlation method.

Stage 3: In the third stage, two two-input multipliers compute  $A_{i,j}^k$  and the product  $B_k C_{i,j}$ .

Stage 4: Finally, a divider computes  $R_k^2(i,j)$  using the results obtained in Stage 3.

Therefore,  $3KL+2$  two-input multipliers, three  $KL$ -input adders and one divider are required to compute  $R_k^2(i,j)$  from  $S_{i,j}$  and  $W_k$ .

### B. Binary Correlation Algorithm

If the windows and search area are preprocessed and transformed to binary images, the correlation algorithm can be implemented using simple digital hardware. Several algorithms for transforming digital images to binary images are given in [5]. When the window  $W_k$  and search area are in binary form, the correlation between  $W_k$  and  $S_{i,j}$  is defined as

$$R_k(i,j) = KL - \sum_{\ell=1}^K \sum_{m=1}^L |S_{i,j}(\ell,m) - W(\ell,m)| \quad , \quad (3-6)$$

for  $1 \leq i \leq M-K+1$  and  $1 \leq j \leq N-L+1$ .

A schematic for the widely used exclusive-OR gate implementation of the binary correlation algorithm given in Equation 3-6 is shown in Figure 3-2. From the schematic, it is clear that the hardware implementation requires  $KL$  exclusive-OR gates and one  $(KL+1)$ -input adder. Detailed description of this binary correlator can be found in [2].

### C. Correlation of Adjacent Pixels Method

An interesting property of the image registration algorithm based on correlation of adjacent pixels is the possibility of computing the feature vector of the subimage  $S_{i,j+1}$  or  $S_{i+1,j}$  from the feature vector of  $S_{i,j}$  with very few arithmetic operations. The above property makes this

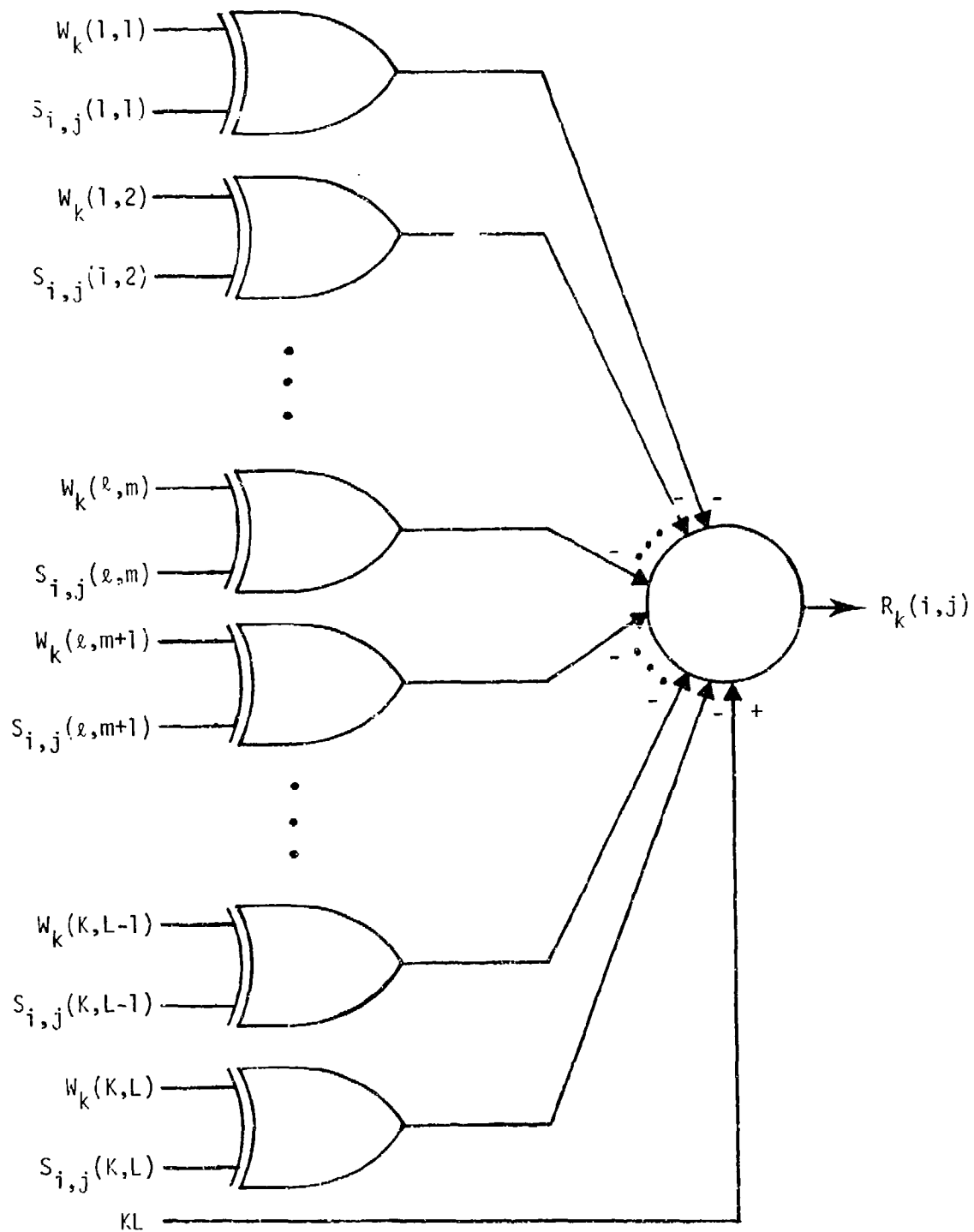


Figure 3-2. Schematic for the binary correlation algorithm.

method extremely suitable for real-time implementation. A schematic for a possible real-time hardware implementation is given in Figures 3-3 through 3-5. It is assumed that the windows and search area have been preprocessed using an algorithm which replaces each pixel by its value minus the average pixel value of the  $K \times L$  array centered about it. This compensates for the difference in d.c. bias between the two imaging systems.

Consider a  $K \times N$  array of shift registers as shown in Figure 3-3 where  $K$  is the number of rows in the reference image and  $N$  is the number of active pixels in each row of the preprocessed search area. At every sampling instant a new pixel of the  $M \times N$  search area enters the array of shift registers at the top left corner as shown in Figure 3-3. During the first sampling period,  $S(1,1)$  enters the shift register  $SR_{1,1}$ . During the second sampling period,  $S(1,1)$  is shifted from  $SR_{1,1}$  to  $SR_{1,2}$  and  $S(1,2)$  enters  $SR_{1,1}$  and so on. Therefore, after  $N$  sampling periods, the first row of shift registers has the  $N$  pixels of the first row of the search area  $S$ . During the  $N+1$ th sampling period,  $S(1,1)$  is shifted from  $SR_{1,N}$  to  $SR_{2,1}$ , the contents of  $SR_{1,i}$  is shifted into  $SR_{1,i+1}$  for  $i = 1, 2, \dots, N-1$  and  $S(2,1)$  enters  $SR_{1,1}$ . The contents of various shift registers at the end of  $KN$  sampling periods is shown in Figure 3-3. During the next sampling period  $S(1,1)$  goes out of the array and  $S(K+1,1)$  enters the array of shift registers and so on. The computation of feature vectors for all  $K \times L$  subimages of the search area as they shift through the array of shift registers is presented next.

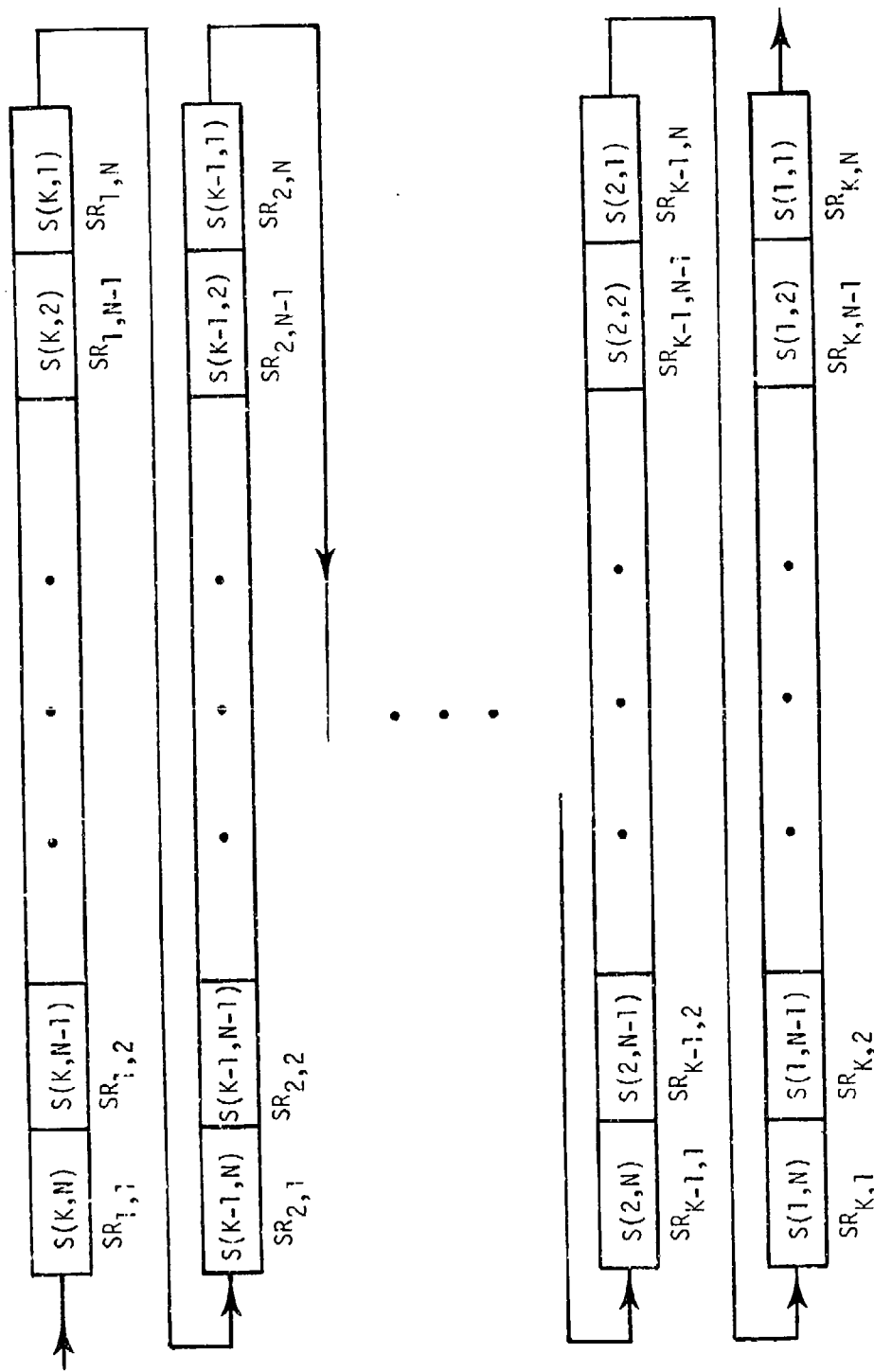


Figure 3-3. KxN array of shift registers containing the first K rows of the search area of size  $M \times N$  at the end of  $KN$  sampling periods.

### 1. Row Feature Extractor

Consider computation of the normalized correlation of adjacent pixels in the first row of the subimage  $S_{1,1}$  using the row feature extractor shown in Figure 3-4. Multipliers M1 and M2, accumulators A1 and A2, and shift registers D1 and D2 are connected to  $SR_{K,N-1}$  and  $SR_{K,N}$  as in Figure 3-4. In order to compute the correlation of adjacent pixels in row 1 of the subimage  $S_{1,1}$ , all shift registers and accumulators shown in Figure 3-4 are first initialized to zero. The search area  $S$  is gated into the  $K \times N$  array of shift registers as described before. At the end of  $KN-1$  sampling periods, the contents of shift registers in row  $K$  of the array are given in Figure 3-5. At this point, M2, D2, and A2 are enabled. It is easy to see that at the end of  $KN-1$  sampling periods  $S^2(1,1)$ , which appears at the output of M2, is added to the previous value in A2 (zero). The content of D2 ( $L$ ), which is zero for this case, is then subtracted from the output of M2. Therefore,

$$Y = S^2(1,1) = S_{1,1}^2(1,1)$$

The shift register D2(1) now contains  $S_{1,1}^2(1,1)$ . Since M1, D1 and A1 are not enabled at the end of  $KN-1$  sampling periods,  $X$  is zero. At this point, M1, D1 and A1 are enabled. At the end of the next sampling period ( $KN$ th sampling period),  $SR_{K,N}$  and  $SR_{K,N-1}$  contain  $S(1,1)$  and  $S(1,2)$ , respectively. Therefore,  $S(1,1) S(1,2)$  appears at the output of M1 and  $S^2(1,2)$  appears at the output of M2. The content of D2( $i$ ) is moved into D2( $i+1$ ) for  $i = 1, 2, \dots, L-1$ , and the output of M2 is gated into D2(1). The content of D2( $L$ ) is shifted out of D2 and the new value in A2 is the old value minus the contents of D2( $L$ ) plus the value at the output of M2. Therefore,



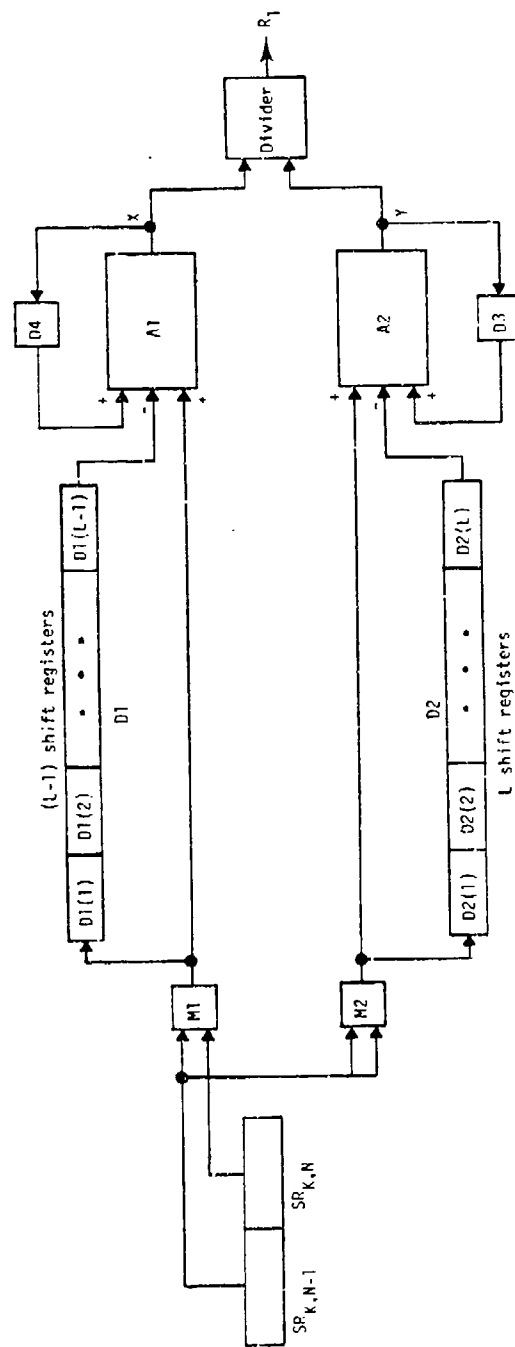


Figure 3-4. Row feature extractor.

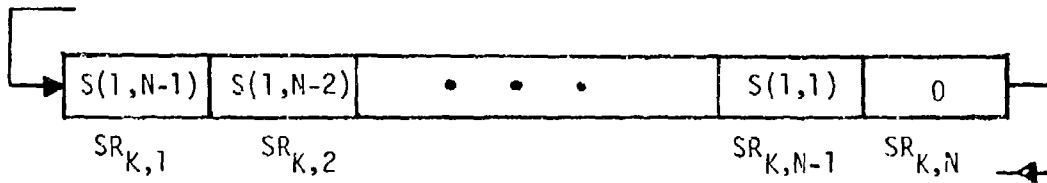


Figure 3-5. Contents of shift registers in row K at the end of  $KN-1$  sampling periods.

$$\begin{aligned} Y &= S^2(1,1) + S^2(1,2) - 0 \\ &= S_{1,1}^2(1,1) + S_{1,1}^2(1,2) \end{aligned} \quad (3-7)$$

Similarly,

$$\begin{aligned} X &= 0 + S(1,1) S(1,2) - 0 \\ &= S_{1,1}(1,1) S_{1,1}(1,2) \end{aligned} \quad (3-8)$$

At the end of  $KN+1$  sampling periods,

$$Y = S^2(1,1) + S^2(1,2) + S^2(1,3) , \quad (3-9)$$

$$\text{and} \quad X = S(1,1) S(1,2) + S(1,2) S(1,3) \quad (3-10)$$

Also,  $D1(1)$  and  $D1(2)$  contain  $S(1,2) S(1,3)$  and  $S(1,1) S(1,2)$ , respectively.

Similarly,  $D2(1)$ ,  $D2(2)$ , and  $D2(3)$  contain  $S^2(1,3)$ ,  $S^2(1,2)$  and  $S^2(1,1)$ , respectively. During every sampling period a product term is added to  $A1$  and a square term is added to  $A2$ . The values in  $D1$  and  $D2$  are shifted left one place and new values enter  $D1(1)$  and  $D2(1)$ . After  $KN+L-2$  sampling periods,

$$X = \sum_{j=1}^{L-1} S(1,j) S(1,j+1) = \sum_{j=1}^{L-1} S_{1,1}(1,j) S_{1,1}(1,j+1) , \quad (3-11)$$

$$Y = \sum_{j=1}^L S^2(1,j) = \sum_{j=1}^L S_{1,1}^2(1,j) , \quad (3-12)$$

$$\text{and } R_1 = \frac{\sum_{j=1}^{L-1} S_{1,1}(1,j) S_{1,1}(1,j+1)}{\sum_{j=1}^L S_{1,1}^2(1,j)} \quad (3-13)$$

From Equation (3-13), it is clear that  $R_1$  is the first component of the feature vector of the subimage  $S_{1,1}$ . At this point, the shift registers  $D2(L)$  and  $D1(L-1)$  contain  $S^2(1,1)$  and  $S(1,1) S(1,2)$ , respectively. During the next sampling period,  $(KN+L-1)$ th sampling period,  $S(1,1) S(1,2)$  shifts out of  $D1(L-1)$  and  $S^2(1,1)$  shifts out of  $D2(L)$ .  $S(1,L) S(1,L+1)$  appears at the output of  $M1$  and  $S^2(1,L+1)$  appears at the output of  $M2$  and therefore,

$$\begin{aligned} X &= \sum_{j=1}^{L-1} S(1,j) S(1,j+1) + S(1,L) S(1,L+1) - S(1,1) S(1,2) \\ &= \sum_{j=1}^{L-1} S_{1,2}(1,j) S_{1,2}(1,j+1) \quad , \end{aligned} \quad (3-14)$$

$$\begin{aligned} Y &= \sum_{j=1}^L S^2(1,j) + S^2(1,L+1) - S^2(1,1) \\ &= \sum_{j=1}^L S_{1,2}^2(1,j) \quad , \end{aligned} \quad (3-15)$$

$$\text{and } R'_1 = \frac{X}{Y} = \frac{\sum_{j=1}^{L-1} S_{1,2}(1,j) S_{1,2}(1,j+1)}{\sum_{j=1}^L S_{1,2}^2(1,j)} \quad (3-16)$$

From Equations (3-16), it is clear that  $R'_1$  is the first component of the feature vector of the subimage  $S_{1,2}$ . Similarly, at the end of the next

sampling period, the first component of the feature vector of  $S_{1,3}$  appears at the output of the divider and so on.

## 2. Column Feature Extractor

Consider computation of the normalized correlation between adjacent pixels in the first column of the subimage  $S_{1,1}$  using the column feature extractor shown in Figure 3-6. The  $SR_{1,N}$  through  $SR_{K,N}$  are the shift registers shown in Figure 3-3.  $M_1, M_2, \dots, M_{K-1}, N_1, N_2, \dots, N_K$  are two-input multipliers. At the end of the  $KN$ th sampling period, the contents of various shift registers of the  $K \times N$  array will be as shown in Figure 3-3. The first column of  $S_{1,1}$  will be in the  $N$ th column of the  $K \times N$  array of shift registers. From Figure 3-6, it is clear that

$$X = \sum_{i=1}^{K-1} S_{1,1}(i,1) S_{1,1}(i+1,1) \quad , \quad (3-17)$$

$$Y = \sum_{i=1}^K S_{1,1}^2(i,1) \quad , \quad (3-18)$$

$$\text{and} \quad C_1 = \frac{X}{Y} = \frac{\sum_{i=1}^{K-1} S_{1,1}(i,1) S_{1,1}(i+1,1)}{\sum_{i=1}^K S_{1,1}^2(i,1)} \quad (3-19)$$

$C_1$  given by Equation (3-19) is the normalized correlation between the adjacent pixels of the first column of  $S_{1,1}$  and is the  $(K+1)$ th component of the feature vector of the subimage  $S_{1,1}$ . During the next sampling period, the second column of  $S_{1,1}$  will move into the  $N$ th column of the  $K \times N$  array of shift registers shown in Figure 3-3 and therefore the new  $C_1$  will be the  $(K+2)$ th component of the feature vector of  $S_{1,1}$ . An efficient

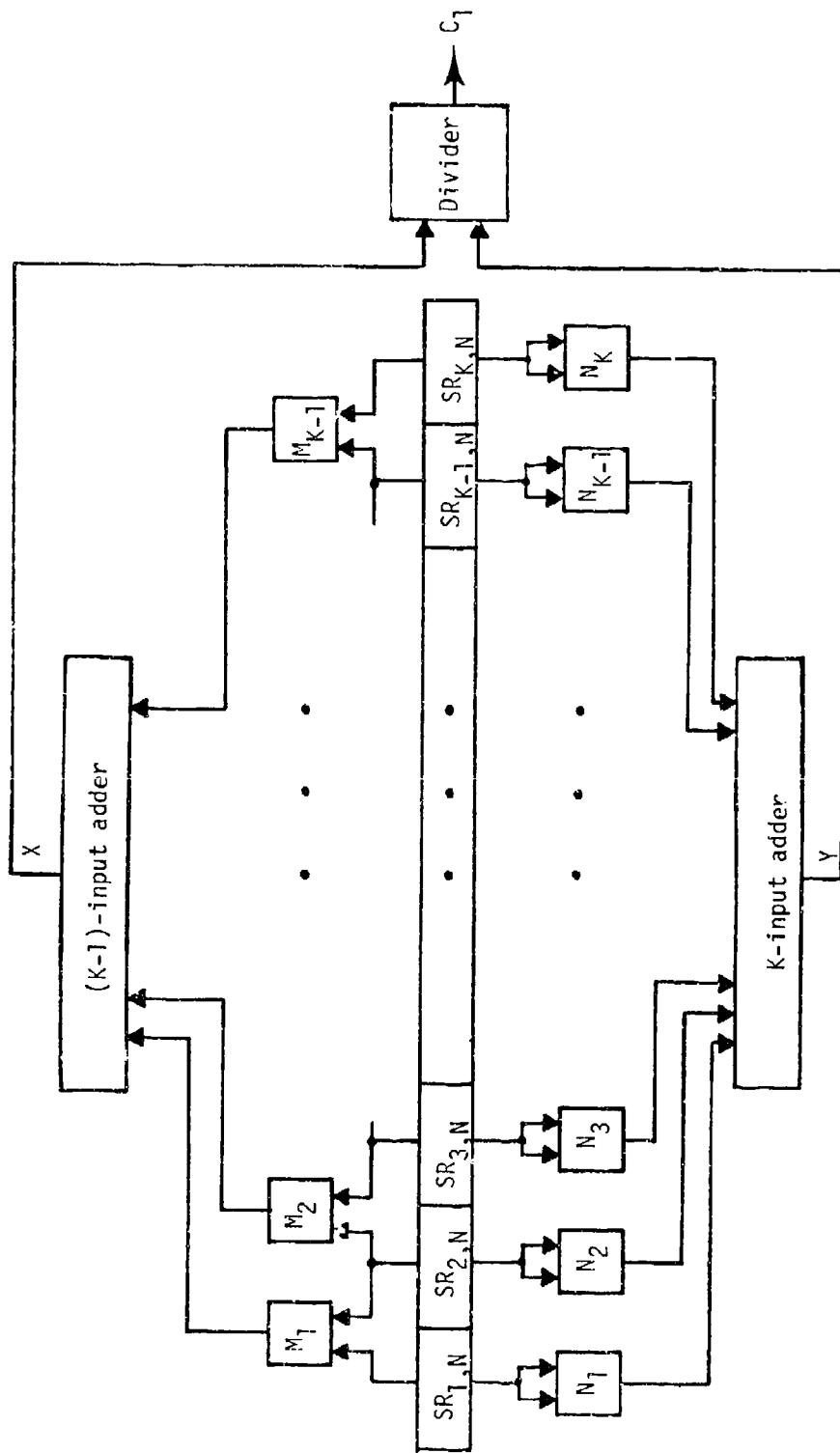


Figure 3-6. Column feature extractor.

and feasible way of implementing the correlation of adjacent pixels method using K row feature extractors and one column feature extractor is presented next.

### 3. Real-time Implementation

The schematic for the real-time hardware implementation of correlation of adjacent pixels method using K row feature extractors and a column feature extractor is given in Figure 3-7. From previous discussion of the operation of feature extractors, it is evident that at the end of the (KN-1)th sampling period,

$$R_i = 0, \text{ for } 1 \leq i \leq K, \quad (3-20)$$

and the correlation of adjacent pixels in the first column of  $S_{1,1}$  appearing at the output of the column feature extractor moves into  $SR_L$ . At the end of KNth sampling period,

$$R_i = \frac{S_{1,1}(i,1) S_{1,1}(i,2)}{S_{1,1}^2(i,1) + S_{1,1}^2(i,2)}, \text{ for } 1 \leq i \leq K \quad (3-21)$$

During the above sampling period, contents of  $SR_{i+1}$  moves into  $SR_i$  for  $1 \leq i \leq L-1$ , contents of  $SR_1$  is lost and the correlation of adjacent pixels in column 2 of  $S_{1,1}$  moves into  $SR_L$ . Finally, at the end of the (KN+L-2)th sampling period,

$$R_i = \frac{\sum_{j=1}^{L-1} S_{1,1}(i,j) S_{1,1}(i,j+1)}{\sum_{j=1}^L S_{1,1}^2(i,j)}, \text{ for } 1 \leq i \leq K, \quad (3-22)$$

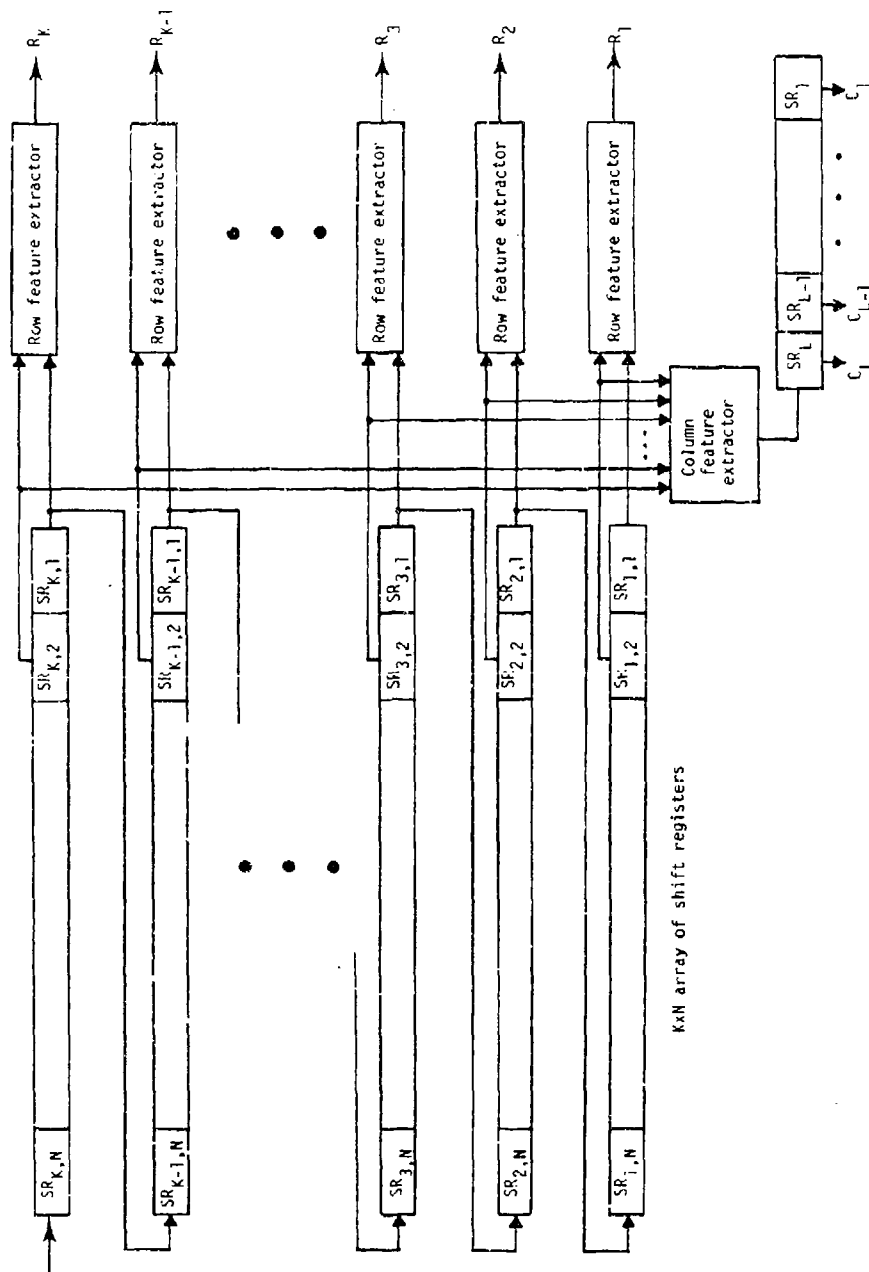


Figure 3-7. Real-time implementation of correlation of adjacent pixels method and the new feature matching method.

$$\text{and } C_j = \frac{\sum_{i=1}^{K-1} S_{1,1}(i,j) S_{1,1}(i+1,j)}{\sum_{i=1}^K S_{1,1}^2(i,j)}, \text{ for } 1 \leq j \leq L \quad (3-23)$$

From Equations (3-22) and (3-23), it is clear that  $[R_1 R_2 \dots R_K C_1 C_2 \dots C_L]^T$  is the feature vector of  $S_{1,1}$ . During the next sampling period,  $[R_1 R_2 \dots R_K C_1 C_2 \dots C_L]^T$  will be the feature vector of  $S_{1,2}$  and so on. In general, at the end of the  $(KN+L+j-3)$ th sampling period  $[R_1 R_2 \dots R_K C_1 C_2 \dots C_L]^T$  will be the feature vector of the subimage  $S_{1,j}$ , for  $1 \leq j \leq N-L+1$ . This completes the computation of feature vectors for subimages  $S_{1,j}$  for  $1 \leq j \leq N-L+1$ . At this point, all the  $K$  row feature extractors and the column feature extractors are re-initialized and the above procedure is repeated to compute the feature vectors of  $S_{2,j}$  for  $1 \leq j \leq N-L+1$ . The above procedure is repeated until feature vectors of all subimages are computed.

#### 4. Matching of Feature Vectors

A possible hardware implementation for the real-time computation of feature vectors of the subimages of the search area has been described in detail. A similar unit consisting of a  $K \times L$  array of shift registers,  $K$  row feature extractors and a column feature extractor, can be used to compute the feature vectors of the  $n$  windows sequentially or  $n$  such units can be used in parallel to compute the  $n$  feature vectors simultaneously. In order to accomplish multiple image registration, the feature vector of each window must be matched with the feature vectors of the subimages of  $S$ . One way of accomplishing the above is shown in Figure 3-8 and is self explanatory.



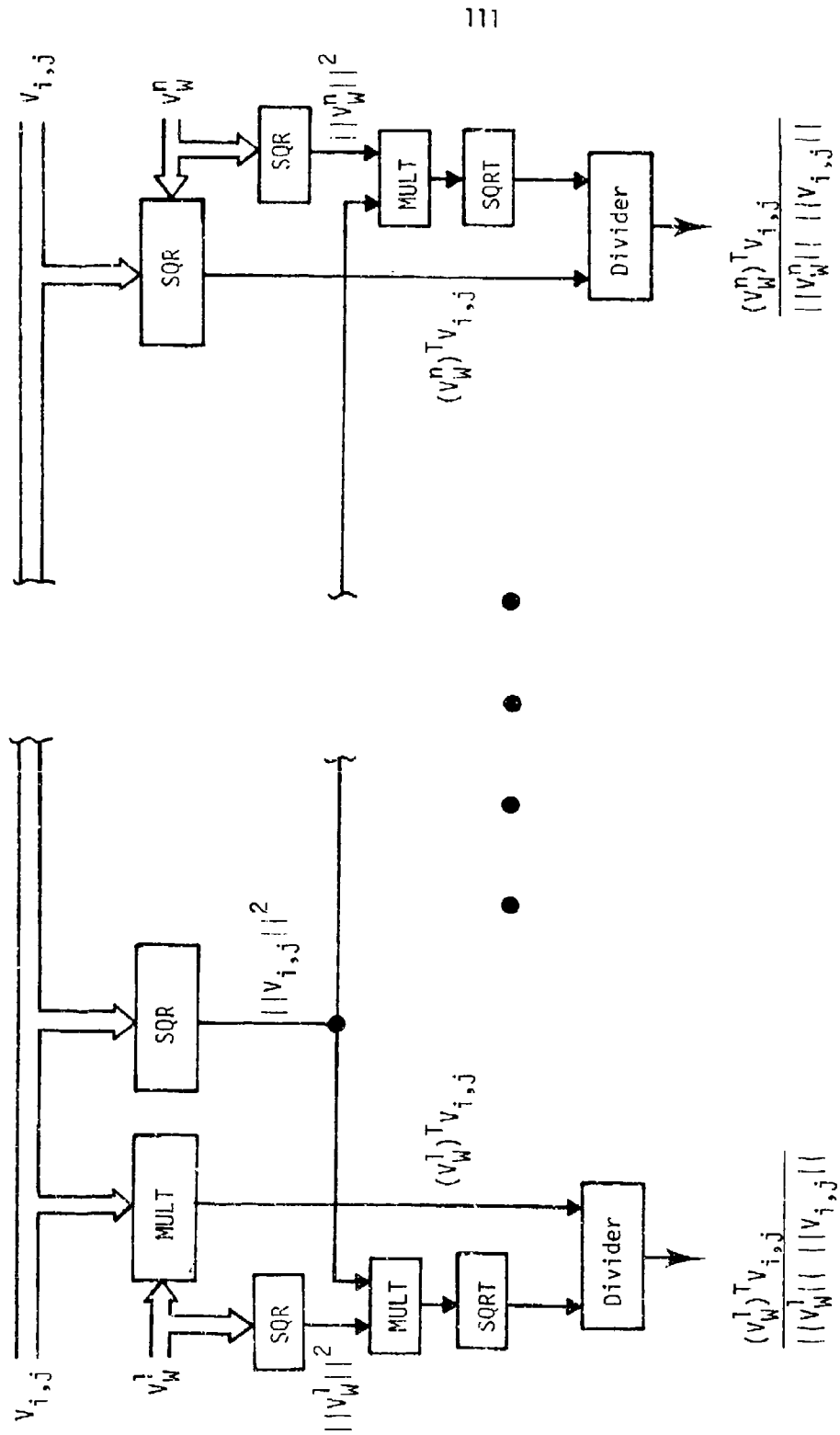


Figure 3-8. Correlation of feature vectors.

#### D. Hardware Requirements of the Standard Correlation and Correlation of Adjacent Pixels Method

A schematic for real-time implementation of the standard correlation algorithm is given in Figure 3-1. For simultaneous registration of  $n$  windows,  $n$  such units must be used in parallel.  $C_{i,j}$  in Equation 3-4 is independent of the windows and need be computed only once. Therefore,  $3KL+2$  two-input multipliers, one divider and three  $KL$ -input adders are required to implement the first correlator. Each additional correlator can be implemented using  $2KL+2$  two-input multipliers, one divider and two  $KL$ -input adders. In other words, simultaneous registration of  $n$  windows using the standard cross correlation method requires  $(2KL+2)n + KL$  two-input multipliers,  $(2n+1)$   $KL$ -input adders and  $n$  dividers.

Each of the  $K$  row feature extractors shown in Figure 3-7 requires 2 multipliers, 2 accumulators, one divider and  $2L+1$  shift registers. The column feature extractor requires  $2K-1$  multipliers, one  $(K-1)$ -input adder, one  $K$  input adder, one divider and  $L$  shift registers. Therefore, the schematic in Figure 3-7 requires a total of  $4K-1$  multipliers,  $2K$  accumulators,  $2KL+K+L$  shift registers (excluding the  $K \times N$  array of shift registers), one  $K$ -input adder, one  $(K-1)$ -input adder and  $K+1$  dividers. The feature vectors of all the  $n$  windows are computed first sequentially, using the hardware shown in Figure 3-7, and then the feature vectors of the subimages of the search area are computed. The matching of feature vectors shown in Figure 3-8 requires,  $\{(2K+2L+1)n + K+L\}$  multipliers,  $n$  dividers  $n$  square root operators, and  $(2n+1)(K+L)$ -input adders. The above results are summarized in Table 16. If the search area and windows are coarsely quantized to two or three levels, multipliers can be implemented using exclusive-

Table 16. Hardware requirements of the standard correlation and correlation of adjacent pixels methods.

	Standard Correlation Method	Correlation of Adjacent Pixels Method
Multipliers	$(2KL+2)n + KL$	$(4K-1) + (2K+2L+1)n + K + L$
Dividers	$n$	$K + 1 + n$
Adders	$2n+1$ $KL$ -input	one $K$ -input one $(K-1)$ -input $2n+1$ $(K+L)$ -input
Shift Registers	$KN + nKL$	$KN + 2KL + K + L$
Square Root Operator	-	$n$
Accumulators	-	$2K$

OR gates and the implementations of the standard correlation algorithm and correlation of adjacent pixel methods become much simpler to realize.

#### E. New Feature Matching Method

In this method, digital image registration is accomplished by using features based on adjacent pixel differences along the rows and columns of a digital image. Once the feature vector of the subimage  $S_{i,j}$  is computed, the feature vector of the next subimage,  $S_{i,j+1}$ , can be computed with very few additional computations. This recursive nature of the computation of feature vectors makes the new feature matching method efficient for real-time hardware implementation. A  $K \times N$  array of shift registers as shown in Figure 3-3 initially containing the first  $K$  rows of the search area of size  $M \times N$  is used for the purpose of extracting row and column features. At each sampling instant, a new pixel from the search area enters the shift register,  $SR_{1,1}$ , pushing the previous content of  $SR_{1,1}$  and the contents of other registers by one place to the right. A possible hardware implementation of the row and column feature extractors is discussed first and a schematic for real-time implementation of the feature matching algorithm is presented next.

##### 1. Row Feature Extractor

A hardware implementation of the row feature extractor is shown in Figure 3-9. The row feature extractor, connected to the shift registers  $SR_{K,N}$  and  $SR_{K,N-1}$  in a  $K \times N$  array of shift registers which stores a part of the search image, is used to compute the row feature element of the first row of the subimage of size  $K \times L$ . Initially the shift registers  $SR_1$ ,

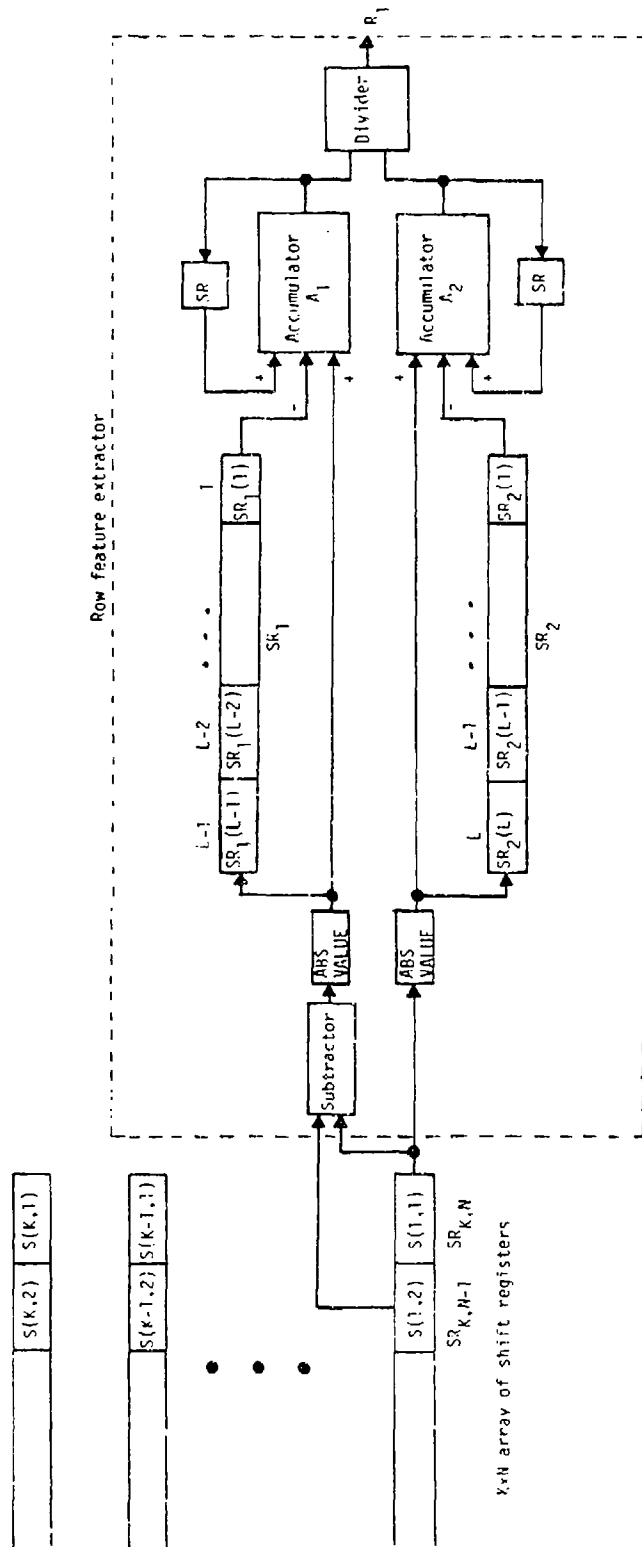


Figure 3-9. Hardware implementation of row-feature extractor.

and  $SR_2$  and the accumulators  $A_1$  and  $A_2$  are cleared. The operation of the row feature extractor is similar in some respects to the one described under correlation of adjacent pixels method and is briefly described below.

During the next sampling instant, the contents of shift registers,  $SR_{K,N}$  and  $SR_{K,N-1}$ , are moved into the subtractor and the content of  $SR_{K,N}$ , which is  $S(1,1)$ , is loaded into the shift register  $SR_2(L)$  and accumulator  $A_2$ . At  $A_2$ , the previous content of  $A_2$  (zero) is added to  $S(1,1)$  and the content of  $SR_2(1)$ , which is also zero, is subtracted from  $S(1,1)$ . At every sampling period the contents of the shift registers in the  $K \times N$  array are shifted to the right by one place, which results in contents of  $SR_{K,N-1}$  occupying  $SR_{K,N}$ . During the next sampling period,  $S(1,2)$  and  $S(1,3)$  are moved into the subtractor and at the same time the output of the subtractor which is the absolute value of the difference between  $S(1,1)$  and  $S(1,2)$  is loaded into  $SR(L-1)$  and accumulator  $A_1$ . At  $A_1$ , the output of the subtractor is added to the previous contents of  $A_1$  and the content of  $SR_1(1)$  is subtracted both of which are zero. Also at this sampling instant,  $S(1,2)$  is moved into  $SR_2(L)$  and  $A_2$  where the previous content of  $A_2$ , which is  $S(1,1)$ , is added to  $S(1,2)$  and the content of  $SR_2(1)$  (zero) is subtracted. After the  $L$ th sampling instant, the quotient of the contents of  $A_1$  and  $A_2$  will give the row-feature element  $R_1(1,1)$  of the first subimage  $S_{1,1}$ , with  $R_1$  appearing at the output of the divider after the  $(L+1)$ th sampling period. The feature element  $R_1(1,1)$  can be expressed in terms of the elements of the first row of the subimage  $S_{1,1}$  according to the equation

$$R_1(1,1) = \frac{N_1(1,1)}{D_1(1,1)} \quad (3-24)$$

where

$$N_1(1,1) = \sum_{j=1}^{L-1} |S_{1,1}(1,j) - S_{1,1}(1,j+1)|, \text{ and}$$

$$D_1(1,1) = \sum_{j=1}^L |S_{1,1}(1,j)|$$

During the  $(L+1)$ th sampling period, the content of  $SR_1(1)$ , which is  $|S(1,1) - S(1,2)|$ , is subtracted from  $A_1$  and  $|S(1,L) - S(1,L+1)|$  is added. At  $A_2$ ,  $S(1,L+1)$  is added to and  $S(1,1)$  is subtracted from the content of  $A_2$ , so that the output of the divider at the  $(L+2)$ th sampling instant is  $R_1(1,2)$ , the row-feature element of the subimage  $S_{1,2}$ , according to the equation

$$R_1(1,2) = \frac{N_1(1,1) - |S_{1,1}(1,1) - S_{1,1}(1,2)| + |S_{1,1}(1,L) - S_{1,1}(1,L+1)|}{D_1(1,1) - |S_{1,1}(1,1)| + |S_{1,1}(1,L+1)|} \quad (3-25)$$

Similarly after  $(L+3)$  sampling periods,  $R_1(1,3)$ , the row feature element of the subimage  $S_{1,3}$  appears at the output of the divider and so on.

## 2. Column Feature Extractor

A possible hardware implementation of the column feature extractor is shown in Figure 3-10. It is again assumed that a part of the search area is stored in the  $K \times N$  array of shift registers shown in Figure 3-3, with the first column of the subimage  $S_{1,1}$  now residing in the last column of the array to which the feature extractor is connected. The adders  $A_1$  through  $A_{K-1}$  compute the adjacent pixel differences of the elements of

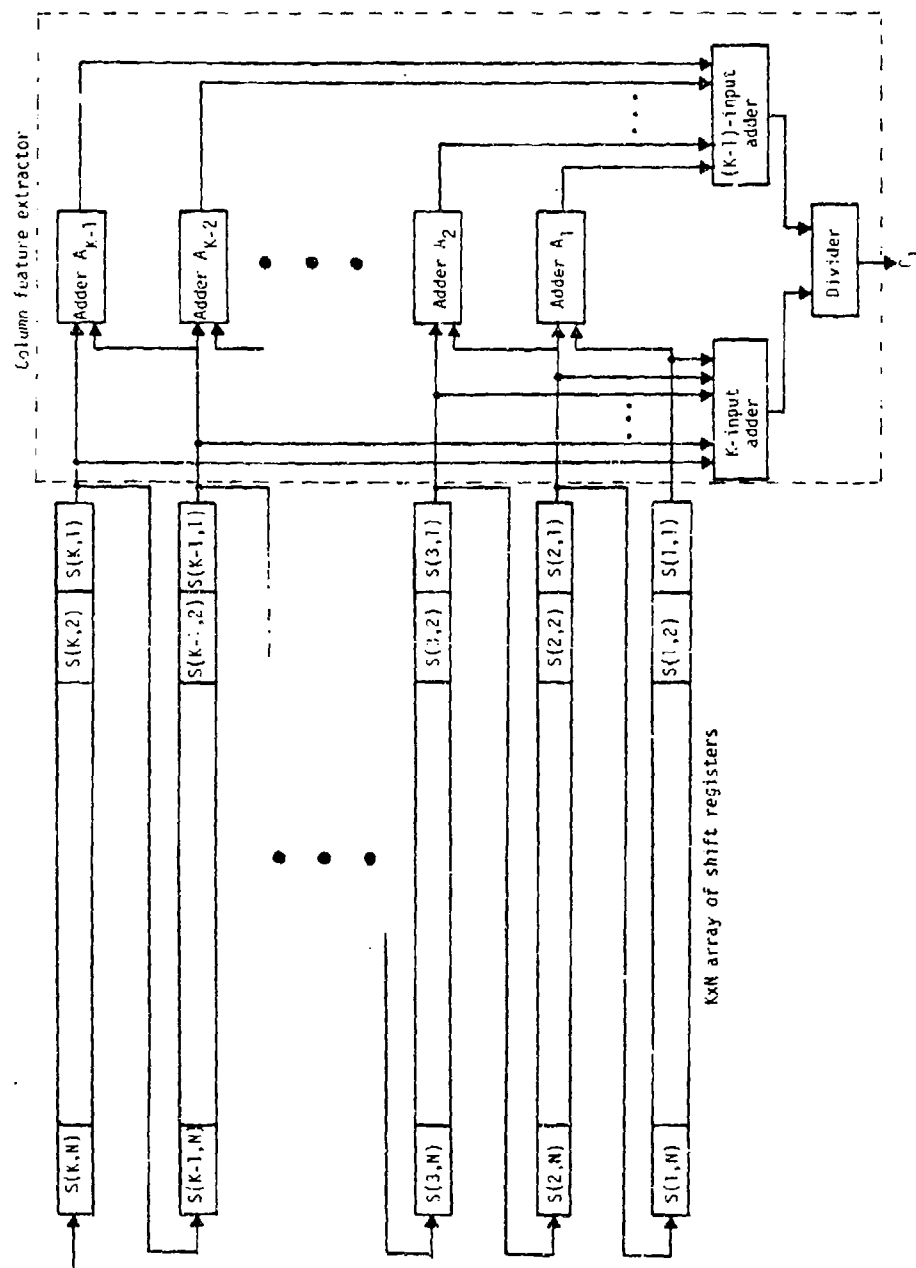


Figure 3-10. Hardware implementation of the column-feature extractor.



the first column whose absolute values are added together by a (K-1)-input adder. A K-input adder shown in Figure 3-10 finds the sum of all the pixels in the first column of the subimage. The outputs of the K-input and (K-1)-input adders are connected to a divider whose output,  $C_1$ , is the column-feature element of the first column of the subimage  $S_{1,1}$  as given by the equation,

$$C_1 = \frac{\sum_{i=1}^{K-1} |S(i,1) - S(i+1,1)|}{\sum_{i=1}^K |S(i,1)|} \quad (3-26)$$

During the next sampling period, the second column of the subimage  $S_{1,1}$  will move into the last column of the  $K \times N$  array of shift registers and the column feature element,  $C_2$ , of the second column of the subimage will be computed by the column feature extractor. The process continues at each succeeding sampling period.

### 3. Real-time Hardware Implementation of the New Feature Matching Method

An efficient way of implementing the new feature matching algorithm using the row and column feature extractors described in the previous sections is shown in Figure 3-7. The K row-feature extractors compute the feature elements of the K rows of the subimage in L sampling periods. The column feature extractor computes the column-feature element of any one column of the subimage in one sampling period, thus computing all the L column-feature elements at the end of L sampling periods. Initially, it takes L sampling periods to compute the (K+L)-dimensional feature vector  $V_{1,1}$  of the subimage  $S_{1,1}$ . Next it takes only one additional sampling

period to compute the feature vector  $V_{1,2}$  of the subimage  $S_{1,2}$  and so on, till all the feature vectors along a row of the search image are computed. The process is repeated to compute the feature vectors along the other rows of the search area, reinitializing (clearing the registers and accumulators) the row-and column-feature extractors every time the computation starts at the beginning of a new row.

#### 4. Multiple Image Registration

Multiple image registration is accomplished by making a decision based on a similarity measure which is expressed in terms of the correlation of the feature vectors of the subimages with the feature vector of each window. A hardware implementation used for this purpose is illustrated in Figure 3-8.

#### 5. Hardware Requirements of the New Feature Matching Method

The complexity of the hardware required for simultaneous registration of  $n$  windows using the real-time hardware shown in Figure 3-7 is given below.

1. A row-feature extractor requires one subtractor (adder), one divider, two accumulators and  $(2L+1)$  shift registers.
2. A column-feature extractor requires  $(K-1)$  adders, one divider, one  $K$ -input adder, one  $(K-1)$ -input adder and  $L$  shift registers.
3. The correlation of feature vectors of subimages and windows for matching purpose (Figure 3-8) requires  $(2K+2L+1)n+K+L$  multipliers,  $n$  dividers,  $n$  square root operators, and  $(2n+1)(K+L)$ -input adders.

The total hardware requirement for the real-time implementation of the feature matching multiple registration algorithm as illustrated in Figure 3-7, with K row-feature extractors, one column-feature extractor and a correlator for matching feature vectors is summarized in Table 17.

#### F. Haar Transform Method

The Haar transform can be computed efficiently using a fast Haar transform (FHT). A flow diagram for computing an 8-point FHT is shown in Figure 2-13. A two-dimensional Haar transform is then computed by repeated application of the one-dimensional FHT. A 2-dimensional transform of a digital image of size  $K \times K$  is obtained by first computing the Haar transform of each of the K rows (horizontal transformation) and then computing the transform of each of the K columns (vertical transformation) of the resulting  $K \times K$  image after horizontal transformation. Some variations of the basic Haar transform, such as the modified Haar transform (MHT) and the rationalized Haar transform (RHT) do not involve multiplication by irrational numbers ( $\sqrt{2}$ ) and can be implemented with adders and subtractors only, making them more attractive for real-time applications. It has also been shown in the previous chapter that RHT performs better than the Haar and the modified Haar transform in registering two images taken from different sensors for the scenes considered. A possible hardware implementation of the two-dimensional Haar transform using both pipelined and parallel one-dimensional Haar transform processing units is discussed in the following sections.

Table 17. Hardware requirements of the new feature matching method.

Adders (subtractors)	$2K-1$	2-input adders
	$2n+1$	$(K+L)$ -input adders
	one	K-input adder
	one	$(K-1)$ -input adder
Multipliers	$(2K+2L+1)n + K + L$	
Dividers	$K + 1 + n$	
Shift registers	$(2L+1)K + L + KN$	
Accumulators	$2K + L$	
Square root operators	$n$	

### 1. Pipelined Implementation of the Haar Transform

Pipelined implementation of the 8-point FHT shown in Figure 3-11 is best understood by referring to the flow diagram shown in Figure 2-13. The pipeline-transform unit shown in the figure is connected to the last row of the  $K \times N$  array of shift registers storing a part ( $K$  rows) of the search area. Initially, the first  $K$  rows of the search area are stored in the array and the Haar transform of the first row of the subimage  $S_{1,1}$  of size  $K \times K$  for  $K = 8$  is computed in the pipeline-transform unit as follows. The pixels of the first row of  $S_{1,1}$  enter the transform unit sequentially one at a time at each sampling instant. The first pixel is stored temporarily in latch  $D_1$  and is presented to the first input  $I_1$  of the adder/subtractor  $AS_1$  of stage 1 during the second sampling instant at which time the second pixel appears at the input of the transform unit and also at the input  $I_2$  of  $AS_1$ . At this point, the sum and difference of the first two pixels are computed, storing the sum temporarily in the shift register  $D_2$ , while the difference is outputted as one of the transform coefficients. The processing is identical at each stage; the sum and difference of successive terms from the preceeding stage are computed; the sum is propagated to the following stage, and the difference is outputted as a transform coefficient. It is to be noted that two shift registers  $D_2$  and  $D_3$  are needed for the second stage and four shift registers  $D_4$  through  $D_7$  are needed for the third stage in order to provide proper inputs to the Adder/Subtractors for the computation of transform coefficients. Initially, a delay of seven sampling periods (neglecting the delay involved in the computation of sum and differences in the adder/subtractors) is needed to get all the 8 transform coefficients of the



first row of the subimage  $S_{1,1}$ . During the next sampling period, all 8 transform coefficients of the first row of the subimage  $S_{1,2}$  are available at the output since the input sequence of pixels of the first row of  $S_{1,2}$  are the input sequence of pixels of the first row of  $S_{1,1}$  shifted one place to the right, and differing only in the last pixel. During the next sampling instant, the coefficients of the first row of the subimage  $S_{1,3}$  is available at the output and so on.

## 2. Parallel Implementation of the Haar Transform

Parallel implementation of the 8-point fast Haar transform using adder/subtractors shown in Figure 3-12 is the hardware realization of the FHT flow diagram shown in Figure 2-13. The operation of the parallel transform unit is easily understood from the flow diagram of the Haar transform. A sequence of eight pixels whose transform is to be computed is input to the transform unit at the same instant of time and the transform-coefficients are available at the output after a small amount of delay occurring in the adder/subtractors. This delay should not exceed one sampling instant for efficient, real-time implementation of the feature matching method using Haar coefficients as discussed in the next section. An increase in speed is achieved by the parallel transform hardware at the expense of an additional four adder/subtractors over that required for the pipeline transform unit shown in Figure 3-11.

## 3. Real-Time Hardware Implementation of the Haar Transform Method

A hardware implementation of the feature matching algorithm based on Haar coefficients using pipelined and parallel Haar transform units described in the previous sections is illustrated in Figure 3-13. The K

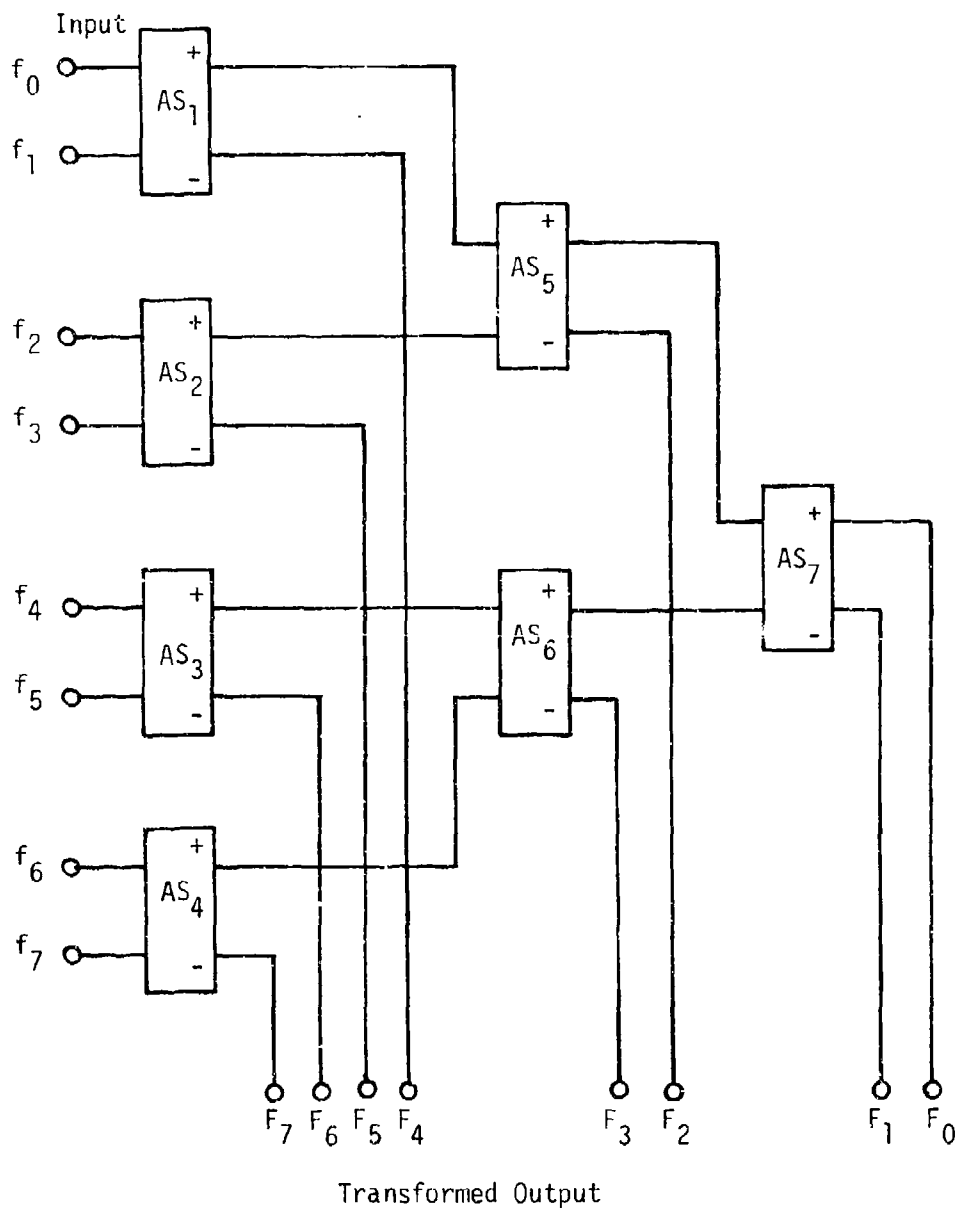


Figure 3-12. Parallel implementation of the 8-point Haar transform.



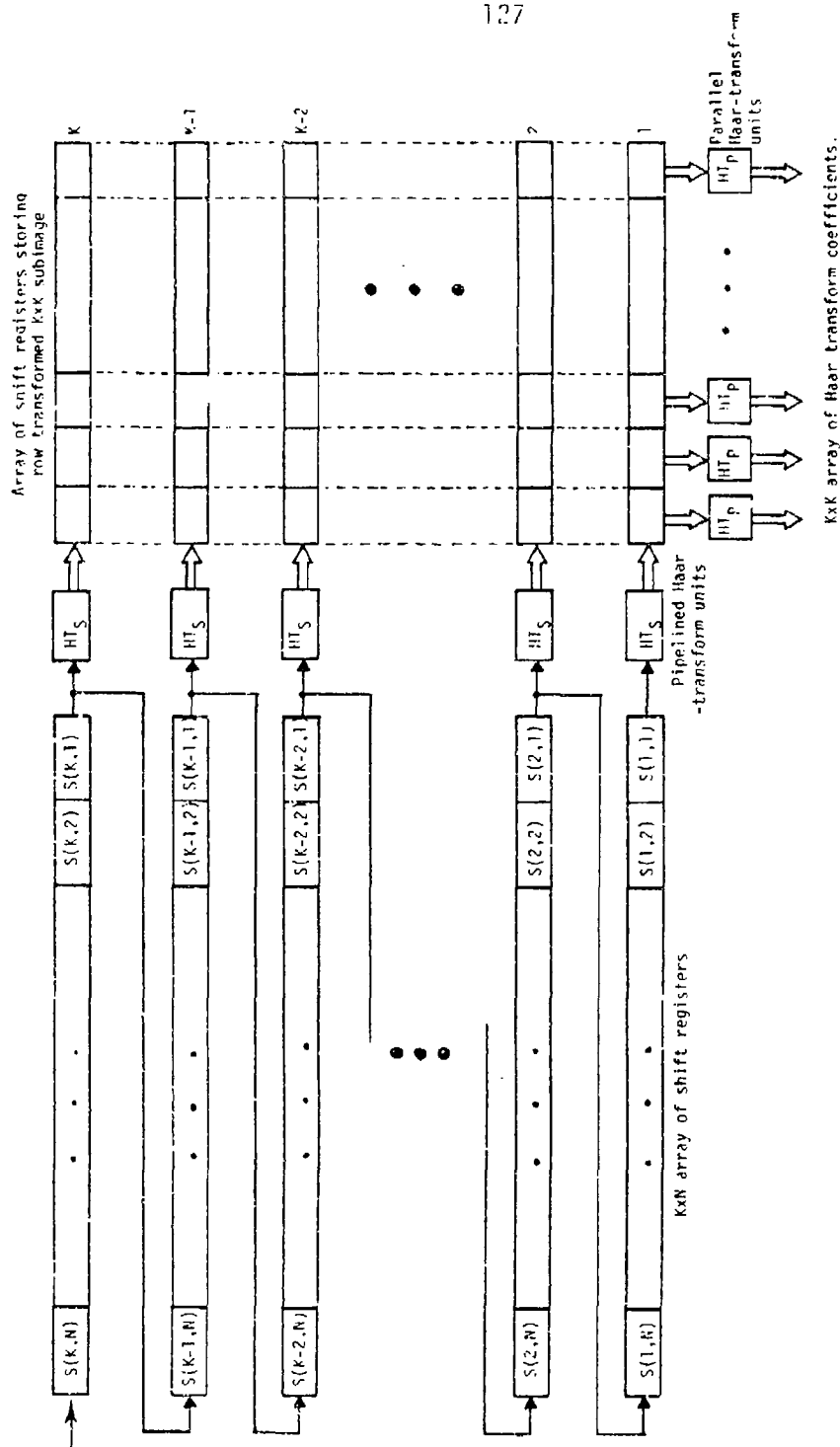


Figure 3-13. Real-time implementation of the feature matching method based on Haar coefficients.

pipelined Haar transform units ( $HT_s$ ) connected to the last column of shift registers of the  $K \times N$  array of shift registers, compute the row transform of the  $K$  rows of the subimages of size  $K \times K$ . Row transformation of the first subimage requires seven sampling instants, with subsequent subimages requiring only one sampling instant. The row transformed subimage is stored in a  $K \times K$  array of shift registers. The  $K$  parallel Haar transform units ( $HT_p$ ) connected to each column of the  $K \times K$  array of shift registers storing the row transformed image, compute the column transform of the image in one sampling instant, thus computing the two-dimensional transform of the original subimage. The  $K \times K$  array of shift registers is updated with a new row transformed subimage at every sampling instant, and the parallel transform units compute the column transform so that the two dimensional transform of the new subimage is available during the next sampling instant. Each time the computation starts at the beginning of a new row of the search image, there is a delay of eight sampling instants before the two dimensional transform of the subimage is computed.

#### 4. Multiple Image Registration

All or a subset of the Haar transform coefficients computed are used to form a feature vector of a subimage or a window. Multiple image registration is then accomplished by making a decision based on a similarity measure which is expressed in terms of the correlation of feature vectors of the subimages with the feature vector of each window. A hardware implementation used for this purpose is shown in Figure 3-8.

### 5. Hardware Requirements of the Haar Transform Method

The amount of hardware required for simultaneous registration on  $n$  windows using the real-time hardware shown in Figure 3-13 is as follows.

1. A pipelined Haar-transform unit computing  $K$ -point FHT, where  $K$  is an integer power of 2 ( $K = 2^P$ ), requires  $P$  adder/subtractors and  $(K-1)$  shift registers, where  $P = \log_2 K$ .
2. A parallel Haar-transform unit requires  $(K-1)$  adder/subtractors.
3. A  $K \times K$  array (i.e.,  $K^2$ ) of shift registers is needed to store the row transformed image.
4. The correlation of  $R$ -dimensional feature vectors of sub-images and windows for matching purposes (Figure 3-8) requires  $(2R+1)n+R$  multipliers,  $n$  dividers,  $n$  square root operators, and  $(2n+1)$   $R$ -input adders.

The dimension  $R$  depends on the number of Haar coefficients selected for matching. The total hardware requirement for real-time implementation of the feature matching method using Haar coefficients with  $K$  pipelined transform units and with  $K$  parallel transform units is summarized in Table 18.

Table 18. Hardware requirements of the feature matching method using Haar coefficients.

Adder/Subtractors	$K[(K-1) + \log_2 K]$
Shift registers	$K(K-1) + K^2 + KN$
Multipliers	$(2R+1)n + R$
Dividers	$n$
Square root operators	$n$
R-input adders	$(2n+1)$

#### IV. HARDWARE vs. TIME TRADEOFFS FOR MULTIPLE TARGET HANDOFF PROBLEM

A number of feature matching algorithms, their image matching accuracies obtained through simulation and suggested hardware implementations, are given in Chapters II and III. In this chapter, several methods of accomplishing multiple target handoff are presented. The trade-off between amount of hardware required and total estimated time for handing off  $n$  targets to  $n$  missile seekers is discussed.

##### A. Method One

Figure 4-1 with correlators 2 through 8 not present is a block diagram of a single target hand-off correlator modified to hand off  $n$  targets sequentially. Initially the helicopter unmask and targets are recognized and cued in the acquisition sensor field of view either manually or using some automatic cueing device such as moving target indicator (MTI) radar. MICOM has demonstrated in its hand-off technology program that manual cueing of targets is too time consuming, exposing the helicopter to hostile fire, thereby reducing the survivability of the aircraft. Target X and Y positions within the acquisition sensor FOV are then loaded into registers in the target assignment circuitry.

As pointed out in Chapter 1, the FOV of the acquisition sensor is assumed to be larger than the FOV of the missile seeker sensors. Also, it is assumed that the LOS of each missile sensor has been prealigned



with the acquisition sensor LOS such that the field of view of each seeker is within the FOV of the acquisition sensor. The next task in the multiple target handoff problem is to locate the missile sensor LOS positions within the acquisition sensor FOV. In method one, this is done sequentially by loading a reference array from the center of the FOV of the first missile seeker into correlator #1 shown in Figure 4-1. The adaptive digitizer, prefiltering, edge detection, and slicing blocks in Figure 4-1 serve the same functions as in a one-to-one handoff correlator. Acquisition sensor video is then preprocessed and used as the live array (search array) by the correlator circuitry. The peak in the correlation surface is the sensor X & Y LOS location within the search area and is stored in a register within the correlator unit. Four additional search area fields are correlated with the reference array and the X & Y LOS locations are loaded into additional registers in the correlator. An algorithm is then used to determine the most probable LOS location or to determine if in fact a valid registration has been achieved. MICOM has demonstrated in its handoff technology program that correlating on five successive fields is sufficient to make the valid match point decision [11].

The valid match point X & Y values are then used by the target assignment circuitry to assign a specific cued target within the acquisition FOV to the missile seeker using a nearest target to sensor LOS rule. Error signals  $\Delta X$ , and  $\Delta Y$ , are then generated and used to cause the missile LOS to slew in the direction of its assigned target. It is assumed that slewing can be accomplished in 0.3 seconds or less. This time is

obviously dependent on the line of sight error magnitude and on the slew rate of the seeker.

Concurrent with the slewing of missile number one LOS the identical procedure described in the above two paragraphs is repeated using video from missile number two sensor. The process is repeated sequentially for up to eight missiles. It is felt that after the first slewing a second look at each missile LOS will be necessary to verify that it is on the right target and that its LOS is close enough to the target for the seeker tracker to lock on. If it is not, then a second slewing will be required.

In computing time required for firing a total of eight missiles, two assumptions have to be made. First, in order for the first missile fired not to blind the last missile fired, it is assumed that the total elapsed time between the first and last missile firing will not exceed 1.75 seconds. Second, in order to allow launch transients to die out on a given missile launcher, firing of two missiles from the same launcher shall not be closer than 0.35 seconds. If two launchers are available and the missiles are fired in tandem from the two launchers, then a missile can be fired every 0.175 seconds. It should be pointed out that these are assumptions with the exact firing time constraints established in a technology program.

Loading of each missile seeker video will require 2 fields (one to allow for proper sync before loading of field of data). Loading of the first acquisition sensor field will also require 2 fields with each four succeeding fields loaded in real-time. Assignment of targets is accomplished by a dedicated processor and done concurrently with correlation



of the following missile video. Therefore, in the computation of total firing time, the assignment of targets is assumed to require no additional time. Slewing of each missile LOS is assumed to be 0.3 seconds or less and is done concurrently with correlation of other missile seekers therefore requiring no added time. The check of each missile LOS to insure proper slewing will require eight fields (2+2+4). Any additional slewing is assumed to be done concurrently with the missile firings. It is assumed that at least one missile will not require any additional LOS slewing and can be fired immediately upon completion of the check.

From the above, the first missile can be fired in  $[8(2+2+4) + 8](1/60) = 1.2$  seconds. All eight missiles can be fired in  $1.2 + 7(0.175) = 2.43$  seconds if all missiles are fired with a minimum time interval.

The advantage of this method is that it requires a minimum amount of hardware change from a single target handoff correlator. The disadvantage is that it requires more time than other methods because of the sequential processing.

Using this method, the total time required to fire six missiles is  $[6(2+2+4) + 8](1/60) + 5(0.175) = 1.81$  seconds and the time required to fire four missiles is  $[4(2+2+4) + 8](1/60) + 3(0.175) = 1.19$  seconds. The firing times for the first missile fired, the firing of eight, six and four missiles are given in Table 4-1 at the end of this chapter for comparison purposes.

#### B. Method Two

The block diagram for method two is shown in Figure 4-1 with all eight correlators present. Targets are cued in the acquisition sensor FOV the same way they were in method one. A reference array from the

center of the first seeker video FOV is loaded into correlator #1. Reference arrays from the remaining seven sensors are then loaded into correlators two through eight sequentially. This will require a maximum of 16 video fields (two fields for each reference to allow for sync errors). The acquisition sensor video is then preprocessed and used as the line (or search) array in all eight correlators in parallel. Again, correlation is performed over five fields in order to insure registration accuracy and reliability. A total of six fields is required for this step. In this step, LOS position results from the five correlation surfaces are averaged after obvious non-correlation fields are thrown out. The X and Y LOS correlation results from all eight sensors are sent to the target assignment circuitry in parallel and targets are assigned to each missile seeker using a minimum total slew distance rule. This step is assumed to require 1/60 second. Error signals are then generated thereby causing each missile LOS to slew. After waiting 0.3 seconds to allow for missile LOS slewing, the above steps are repeated. At this point, it is assumed that several missile LOS are aligned such that their trackers can lock on to their target and firing can occur. Any additional slewing of missiles which are not aligned to the desired accuracy is assumed to be done while the first missiles are being fired and therefore does not require additional time.

From the above, the first missile can be fired in  $2(16+6+1)(1/60) + 0.3 = 1.07$  seconds and all eight missiles can be fired in  $1.07 + 7(0.175) = 2.29$  seconds. The advantage of this method is that it offers a slight total firing time improvement. The disadvantage is that eight full correlators are required.

For comparison with method one, six missiles can be fired in  $2(12+6+1)(1/60) + 0.3 + 5 (0.175) = 1.81$  seconds and four missiles can be fired in  $2(8+6+1)(1/60) + 0.3 + 3 (0.175) = 1.33$  seconds.

### C. Method Three

The block diagram for method three is shown in Figure 4-1 with only correlators 1 through 4 present. Targets are cued as in method one and reference arrays from four seeker videos are loaded sequentially in correlators 1 through 4. This will require a maximum of eight video fields. The acquisition sensor video is then preprocessed and five fields are correlated as in method two (six fields of time are required). Target assignment and error signal generation occurs concurrently with loading the following four reference arrays and therefore requires no additional time. After the correlation surfaces for missile video five through eight are processed and an additional four field delay is added to allow the full 0.3 seconds (or 18 equivalent fields) slewing time, the first four missile LOS are recomputed to check on the slewing accuracy. It is assumed at this point that at least two of these four LOS are within tracker lock-on accuracy and can be fired. Error signals for the others will be generated and a second slewing will occur concurrent with checking the slewing accuracy for missiles five through eight. From this point on, all calculations can be accomplished within the missile firing times and therefore contribute no additional time to the total.

From the description of method three in the above paragraph, the first missile can be fired in  $[3(8+6) + 4](1/60) = 0.77$  seconds and all eight missiles can be fired in  $0.77 + 7 (0.175) = 2.00$  seconds. The reason this method is faster than method two is that missile LOS slewing of

four missiles is occurring simultaneously with calculation of the LOS of the other four missiles thereby reducing the total firing time. By matching three missile video LOS at a time, six missiles can be fired in  $[3(6+6) + 6](1/60) + 5(.175) = 1.58$  seconds. By matching four missile video LOS concurrently, four missiles can be fired in  $2(8+6)(1/60) + 0.3 + 3(.175) = 1.29$  seconds. By matching only two missile videos concurrently then slewing these two while the remaining two are matched, four missiles can be fired in  $[3(4+6) + 8](1/60) + 3(.175) = 1.16$  seconds.

The advantage of this method is that it performs faster than either method 1 or 2 and requires only one-half the number of correlators as method two.

#### D. Method Four

Method four employs three correlators which operate in parallel. When firing on eight targets, the correlators match 3+3+2 missile videos. The first missile can be fired in  $[(6+6) + (6+6) + (4+6) + (3+6)](1/60) = 0.77$  seconds and all eight missiles can be fired in  $0.77 + 7(.175) = 2.0$  seconds. Six missiles can be fired in  $[3(6+6) + 6](1/60) + 5(.175) = 1.58$  seconds. By correlating two missile videos simultaneously, four missiles can be fired in 1.16 seconds, the same as method three.

This method yields identical firing time results to method three but requires one less correlator array. Another advantage of this method is that a slewing time of over 0.5 seconds is available for the first missile LOS slewing when eight targets have been cued.

## E. Method Five

Method five employs only two correlators which operate in parallel. The first missile can be fired in  $5(4+6)(1/60) = 0.83$  seconds and all eight missiles can be fired in  $0.83 + 7 (.175) = 2.06$  seconds. The LOS slewing time for the first missile is 0.5 seconds. Six missiles can be fired in  $4(4+6)(1/60) + 5(.175) = 1.54$  seconds and four missiles can be fired in  $[3(4+6) + 8] (1/60) + 3 (.175) = 1.16$  seconds.

## F. Method Six

In method six, one of the feature matching algorithms discussed in Chapters 2 and 3 is used for multitarget hand-off. As shown in Figure 4-2, targets are cued in the same manner as for all previous methods. The adaptive digitizer, prefiltering, and edge detection blocks are identical to the previous methods. The segmentation block segments the scene into target and background and the feature extractor computes the feature vector of the scene. The target assignment circuit is identical to previous methods.

The first step is to extract features from the missile video reference scenes sequentially and store in the feature comparison circuit. Because the missile video is not synchronized, two fields per reference are required for this step. Features are then computed for all possible reference size subarrays within the live scene (acquisition sensor video) in real time and used by the feature comparator circuit to find the match point.

If one feature comparator circuit is used, the time before the first missile is fired and the total time to fire all eight, six or four



missiles are identical to the times for method one. With eight, four, three, or two feature comparitors the times are identical to those for methods two through five respectively. Therefore, the feature matching methods do not offer a time advantage over the standard correlation method. The advantage of this method is a reduced amount of hardware required if the segmentation and feature extraction circuits are implemented in very large scale integrated circuits (VLSIC) as the correlator circuits are presently implemented. Less hardware should be required because of the smaller amount of multiplications, additions and subtractions required as shown in previous chapters. If these VLSIC are not presently available, this method involves more of a technology risk. Development of these VLSIC chips, however, would be a step toward building an automatic target recognizer as explained in method seven.

#### G. Method Seven

Figure 4-3 is a block diagram of the functions required for an automatic target recognizer. Except for the target classifier, which distinguishes a tank from a truck or an APC, etc., all of the blocks appear in the block diagram shown in Figure 4-2 of a feature matching multitarget handoff device. One difference, which is not apparent in the block diagrams, is that significantly more feature information is required for target classification than for image matching. Also, target classification usually requires working with frames rather than fields of video in order to recognize at the desired ranges. Developing a feature matching multitarget handoff device shown in Figure 4-2 would therefore be a step toward recognition and handoff as shown in Figure 4-4.

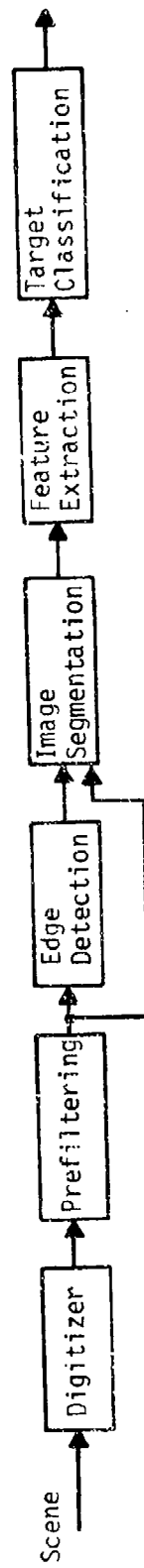


Figure 4-3. Automatic target recognizer block diagram.



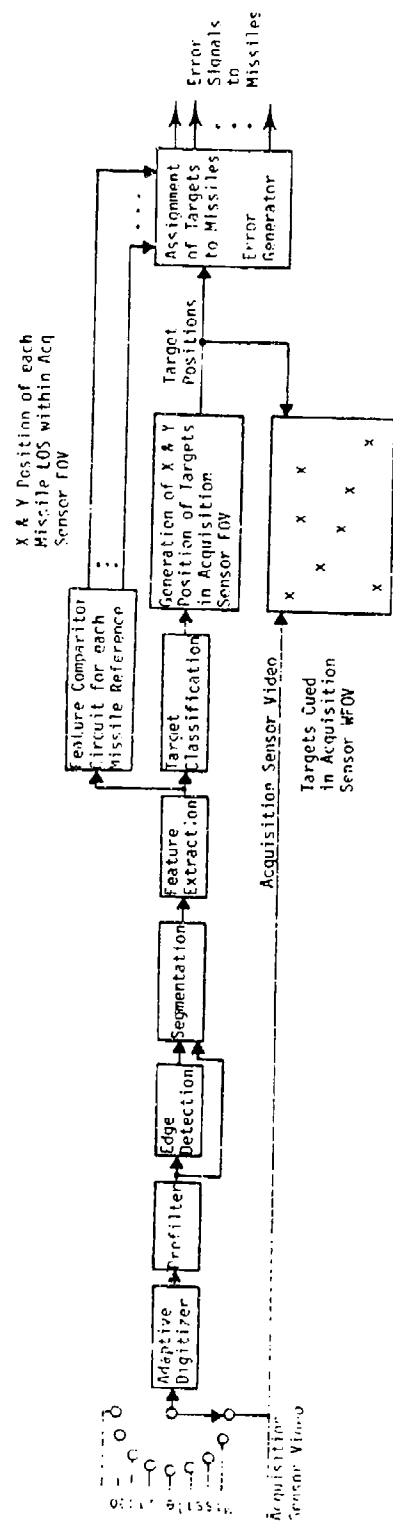


Figure 4-4. Block diagram of combined target recognizer and feature matching multitarget handoff hardware.

The system shown in Figure 4-4 is initially operated in the target recognition mode. Up to eight targets are identified and their X & Y positions within the acquisition sensor FOV are sent to the target assignment and error generation circuitry and to the gunner's monitor where symbology is used to identify targets. Video from each missile sensor is then processed sequentially generating features of reference arrays taken from the center of the FOV. It is anticipated that a subset of the features used for target recognition will be used for image matching. The prefiltering, edge detection, segmentation, and feature extraction can all be implemented in pipelined very large scale integrated circuits. Therefore, a small delay occurs before the first features are generated with the remainder generated in real time. After scaling and prefiltering, features are then generated for all possible KxL subarrays of the acquisition sensor video, where KxL is the size of the reference arrays. These features are then used to find the best match of each missile reference within the acquisition sensor FOV.

With only one feature comparison circuit, the time for firing one, eight, six, or four missiles is identical to the time for method one. With eight, four, three, or two feature comparison circuits, the times are identical to methods 2, 3, 4, or 5, respectively. Since the time lines for feature matching are the same as for the image matching methods, the main advantage of the feature matching method is less hardware requirement. The lesser hardware is a result of the smaller amount of equivalent additions required to generate the  $n$  correlation surfaces.

Table 19. Firing times for seven methods.

	Firing of 1st Missile of Eight	8 Missiles	6 Missiles	4 Missiles
Method #1 1 Correlator	1.2	2.43	1.81	1.19
Method #2 8 Correlators	1.07	2.29	1.81	1.33
Method #3 4 Correlators	0.77	2.00	1.58	1.16
Method #4 3 Correlators	0.77	2.00	1.58	1.16
Method #5 2 Correlators	0.83	2.06	1.54	1.16
Method #6 & #7	With 1, 8, 4, 3, or 2 feature comparison circuits times are equivalent to Methods 1, 2, 3, 4, or 5 respectively			

## V. CONCLUSIONS AND RECOMMENDATIONS

This chapter delineates the major conclusions and recommendations resulting from the work on this contract. Most of the conclusions and recommendations presented in this chapter have been given and justified within the first four chapters of this report.

### A. Conclusions

1. The accuracy and reliability of several feature matching digital image registration algorithms were investigated through simulation using typical military type scenes. Because of previous analysis and hardware design, a binary correlator (exclusive-OR gate implementation) was used as a basis for comparison of the performance of feature matching algorithms in this report. When reference images were chosen to include prominent features in the reduced high resolution FLIR images, the performance of the binary correlation algorithm was satisfactory. When each reference image was chosen from the center of the field of view of its corresponding high resolution image, the method's performance was not as good.
2. The moments method and the method based on intraset and interset distances were shown not to be accurate for the type of imagery considered in this work.

3. The registration method based on correlation of adjacent pixels was simulated with and without intensity level normalization of images. When images were used without intensity level normalization, the method failed for all sixteen pairs of scenes even when the reference images were chosen to include the prominent features of the reduced high resolution images. However, intensity level normalization significantly improved the performance of the method indicating the necessity of some kind of normalization to compensate for the difference in the d.c. gains of the imaging systems.
4. A preprocessing algorithm which replaces each pixel by its value minus the average pixel value of the  $K1 \times L1$  array centered about it compensates for the difference between d.c. gains of the two imaging systems and yet allows the computation of  $V_{i,j+1}$  or  $V_{i+1,j}$  from  $V_{i,j}$ . The performance of the method using preprocessed images is comparable to that of the binary correlation algorithm.
5. The performance of the method based on correlation of adjacent rows and columns was better than that of the method based on correlation of adjacent pixels in rows and columns. This is especially true when the reference images have prominent horizontal and vertical edges. A schematic for a possible real-time implementation of the method of correlation of adjacent pixels was given in Chapter III. This implementation requires less hardware than that of the standard cross correlation algorithm.

6. In order to reduce the hardware requirements further, a new method based on the summation of the absolute values of the differences between adjacent pixels on the rows and on the columns was proposed. Simulation results indicate that the performance of this method is comparable to the correlation of adjacent pixels method discussed in conclusions 3, 4, and 5.
7. Fast implementations of the Haar, modified Haar, and rationalized Haar transforms were presented. Simulations showed that these methods performed comparably with the correlation of adjacent pixel methods. The rationalized Haar transform is easier to implement in hardware and gave better results than the other Haar methods.
8. Suggested hardware implementations for the most promising methods are given in Chapter III. The absolute value of adjacent pixel difference method and the rationalized Haar transform method were the simplest to implement in hardware.
9. Seven methods for accomplishing automatic handoff of multiple targets were presented in Chapter IV. Table 4-1 gives the minimum time required to fire the first missile and the minimum time required to fire eight, six, or four missiles in tandem from two launchers. From Table 4-1 it is clear that there is no time advantage to finding the missile lines of sight using one of the feature matching techniques. The advantage of the feature matching techniques is less hardware requirements due to the smaller amount of equivalent additions needed for multiple target handoff.

The lesser hardware requirement, however, is conditional on implementing the feature extraction algorithms in LSI circuits.

#### B. Recommendations

1. If the desired end result of the current technology program is to build and demonstrate a multiple target handoff device, it is recommended that this device be an upgraded version of the single target handoff correlator with two correlators operating in parallel. Some additional circuitry will be required to handle the sync problems and some additional software will be required. The single target handoff device has been tested and shown to work. Upgrading this device to have two correlators operating in parallel is judged not to be a significant technology risk.
2. If, however, the objective of the current technology program is to reduce the overall helicopter exposure time, some method of automatically cuing potential targets in the precision pointing and tracking system video must be provided. Moving target indicator (MTI) radar or an automatic target recognizer (or both) is needed. It is well known that automatic target recognizers compare features of possible targets with stored features of desired targets. Therefore, developing a multiple target handoff device using a feature matching technique would be a step toward incorporating the automatic target recognition and handoff within the same hardware device. It should be pointed out that the feature matching multiple target handoff technique involves a greater technology risk than the two parallel correlator technique.

Based on the results of the work on this contract, if feature matching is used to accomplish multiple target handoff, the absolute value of adjacent pixel difference method or the rationalized Haar transform method should be used. To upgrade the system to accomplish automatic target recognition as well as target handoff a number of additional target/background features will be required.



## REFERENCES

- [1] J. S. Boland, III, and H. S. Ranganath, "Automatic handoff of multiple targets", Final Technical Report-Phase I, Contract DAAH01-80-C-0258, U.S. Army Missile Command, Redstone Arsenal, AL, September 30, 1980.
- [2] J. S. Boland, III, L. J. Pinson, G. R. Kane, M. A. Honnell, and E. G. Peters, "Automatic target hand-off using correlation techniques", Final Technical Report, Contract DAAH-1-76-0396, U.S. Army Missile Command, Redstone Arsenal, AL, January 31, 1977.
- [3] J. S. Boland, III, and H. S. Ranganath, "Correlation algorithm development", Final Technical Report, Contract DAAK40-79-M-0016, U.S. Army Missile Research and Development Command, Redstone Arsenal, AL, April 30, 1979.
- [4] J. S. Boland, III, and H. S. Ranganath, "Quantization effects on target hand-off from TV-to-IR digitized scenes", Final Technical Report, Contract DAAK40-79-M-0104, U.S. Army Missile Command, Redstone Arsenal, AL, September 15, 1979.
- [5] J. S. Boland, III, L. J. Pinson, and E. G. Peters, "Automatic target hand-off for non-compatible imaging systems", Final Technical Report, Contract DAAK40-77-C-0156, U.S. Army Missile Research and Development Command, Redstone Arsenal, AL, September 15, 1978.
- [6] E. L. Hall, Computer Image Processing and Recognition, Academic Press, New York, 1979.
- [7] V. V. Dixit, "Edge extraction through Haar transform", Fourteenth Asilomar Conf. on Circuits, Systems and Computers, pp. 141-143, November 1980.
- [8] R. T. Lynch, and J. J. Reis, "Haar transform image coding", 1976 National Telecom. Conf., Conf. Record, pp. 44.3-1 to 44.3-5.
- [9] H. F. Harmuth, Sequency Theory-Foundations and Applications, Academic Press, New York, 1977.
- [10] R. C. Gonzalez, and P. Wintz, Digital Image Processing, Addison-Wesley Pub. Co., Inc., Mass., 1977.

- [11] John Herring and P. H. McIngvale, "Test Program for Automatic Target Handoff Correlator TV-IR and IR-IR Preprocessing Algorithms", MICOM Technical Report TR-T-79-87, September 1979.

## DISTRIBUTION

## No of Copies

Program Manager Advanced Attack Helicopter  
ATTN: DRCPM-AAH, MG E. Browne  
4300 Goodfellow Blvd  
St Louis, MO 63120

1

Project Manager  
TADS/PNVS  
ATTN: DRCPM-AAH-TP, COL D. Wray  
S. Smith  
4300 Goodfellow Blvd  
St Louis, MO 63120

1

1

Commander  
US Army Aviation Systems Command  
ATTN: DRSAB-EV, D. Weller  
DRDAV-NC, T. Myers  
4300 Goodfellow Blvd  
St Louis, MO 63120

1

1

Commander  
US Army Aviation Center  
ATTN: ATZQ-TSM-S  
ATZQ-TSM-A  
ATZQ-D-MS  
Ft Rucker, AL 36360

1

1

1

Commander  
Headquarters TCATA  
ATTN: ATCAT-CAD-TA  
Ft Hood, TX 76544

1

Commander  
Headquarters TRADOC  
ATTN: ATCD-CM  
Ft Monroe, VA 23351

1

Program Manager  
Advanced Scout Helicopter  
ATTN: M. Jackson  
4300 Goodfellow Blvd  
St Louis, MO 63120

1

IIT Research Institute  
ATTN: GACIAC  
10 West 35th Street  
Chicago, IL 60616

1

## DISTRIBUTION (cont'd)

## No of Copies

Defense Documentation Center Cameron Station Alexandria, VA 22314	12
DRCPM-HD	1
DRCPM-HDE, ATTN: J. Service	1
DRSMI-X	1
DRSMI-R, Dr. McCorkle	1
RG	1
RGC	15
RGT	1
REO	1
REI	1
RE	3
RR, Dr. Hartman	1
RR, Dr. Guenther	1
RR, Dr. Gamble	1
RN, Mr. Dobbins	1
DRSMI-LP, Mr. Voigt	1
DRSMI-V/J. Keats	1
Commander US Air Force Armament Laboratory Eglin AFB, FL 32 2	1
Director Defense Advanced Research Projects Agency ATTN: Dr. James Tegnalia 1400 Wilson Blvd Arlington, VA 22209	1
Science and Technology Division Institute of Defense Analysis ATTN: Dr. Vincent J. Corcoran 400 Army-Navy Drive Arlington, VA 22202	1
ASD/RWNM ATTN: Mr. E. Wallace Wright Patterson AFB, OH 45433	2
Director Office of Missile Electronic Warfare ATTN: DELEW-M-ST, Mr. Jim Hardison White Sands Missile Range, NM 88002	1

## DISTRIBUTION (cont'd)

No of Copies

Commander US Army Armament R&D Command  
Picatinny Arsenal

ATTN: Dr. Gyorg  
DRDAR-FCD-W

Dover, NJ 07801

1

1

Commander US Army Electronics R&D Command

ATTN: DRSEL-TL-I, Dr. Jacobs

DRSEL-CT, Dr. R. Buser

DELET-M, Mr. Walt Gelnovatch

Mr. N. Wilson

Ft Monmouth, NJ 07703

1

1

1

1

Commander

US Army Electronics Command

NV&EOL

ATTN: DELNV-SI, John Johnson

Ft Belvoir, VA 22060

1

Commander

US Army Training and Doctrine Command

Ft Monroe, VA 23341

1

Commander

US Army Combined Arms Combat Development Activity

Ft Leavenworth, Kansas 66027

1

Commander

US Army Armor Center

Directorate for Armor Aviation

ATTN: ATSB-AADO-MS, LTC Don Smart

Ft Knox, KY 40121

1

Headquarters

Department of the Army

ATTN: DAMA, WSM, MAJ Belch

Washington, DC 20310

1

DRSMI-IYB, Mr. Ingram

1

Texas Instruments

ATTN: Dr. Jack Pridgen

PO Box 6015

Dallas, TX 75222

1

Goodyear Aerospace Corporation

ATTN: H. Pfeiffer

Akron, Ohio 44315

2

DISTRIBUTION (cont'd)

No of Copies

Lockheed Missiles and Space Company  
ATTN: Dr. James Pearson  
3251 Hanover Street  
Palo Alto, CA 94304

1

Martin Marietta Aerospace  
ATTN: Charles R. Layne MP 142  
Max W. Farrow, MP 544  
PO Box 5837  
Orlando, FL 32855

1

1

RCA Government Systems Division  
ATTN: Richard Foley  
Camden, NJ 08102

1

TASC  
ATTN: Martin Svedlow  
6 Jacob Way  
Reading, MA 01867

1

Auburn University  
ATTN: Dr. J. S. Boland, III  
Electrical Engineering Department  
Auburn University, AL 36849

10

U. S. Army Aviation R&D Command  
Applied Technology Laboratory  
ATTN: Mr. Nick Kailos  
DAVDL-ATL-ASW  
Ft Eustis, VA 23604

1